Universidade Federal do Paraná

Hegler Correa Tissot

Normalisation of imprecise temporal expressions extracted from text

Curitiba PR
2016

Hegler Correa Tissot

Normalisation of imprecise temporal expressions extracted from text

Final Doctoral Thesis presented as partial requirement to obtain a Ph.D. in Computer Science at the Federal University of Paraná (UFPR).

Area: *Computer Science*.

Supervisor: Marcos Didonet Del Fabro.

Co-supervisor: Angus Roberts.

Curitiba PR
2016

Tissot, Hegler Correa
Normalisation of imprecise temporal expressions extracted from text /
Hegler Correa Tissot. – Curitiba, 2016
127 f. : il.

Tese (doutorado) – Universidade Federal do Paraná, Setor de Ciências Exatas,
Programa de Pós-Graduação em Informática.
Orientador: Marcos Didonet Del Fabro
Coorientador: Angus Roberts

1. Tecnologia da informação. 2. Palavras-Chave. I. Del Fabro,
Marcos Didonet. II. Roberts, Angus. III. Título

CDD 004.678

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós Graduação em INFORMÁTICA
Código CAPES: 40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Tese de Doutorado de **HEGLER CORREA TISSOT**, intitulada: "**Normalisation of imprecise temporal expressions extracted from text.**", após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação.

Curitiba, 05 de Abril de 2016.

Prof MARCOS DIDONET DEL FABRO (UFPR)
(Presidente da Banca Examinadora)

Prof ANGUS ROBERTS (SHEFF)

Prof EDUARDO CUNHA DE ALMEIDA (UFPR)

Prof JULIO CÉSAR NIEVOLA (PUC/PR)

Prof LUIZ EDUARDO SOARES DE OLIVEIRA (UFPR)

Acknoledgements

I would like to thank:

My supervisor, Marcos Didonet Del Fabro, and my co-supervisor, Angus Roberts (The University of Sheffield, UK), for all the provided support to help me in this work.

My colleague, Genevieve Gorrell (The University of Sheffield, UK), for helping in the Clinical TempEval with the development of the SVM approach we submitted to be evaluated in that task.

My colleague, Leon Derczynski (The University of Sheffield, UK), for his collaboration about temporal information extraction.

My friend, Suzanne (Sue) Duggan, for helping me on reviewing the final version of my thesis.

The NLP group and specially the GATE group from The University of Sheffield, UK.

The Mayo Clinic for the permission to use the THYME corpus.

The *InfoSaude* team for the permission to use the *InfoSaude* corpus.

CAPES, which financed this work.

The National Institute for Health Research (NIHR) Biomedical Research Centre and Dementia Biomedical Research Unit at South London and Maudsley NHS Foundation Trust and King's College London.

Everyone who helped me in different ways to conclude this work.

Resumo

Técnicas e sistemas de extração de informações são capazes de lidar com a crescente quantidade de dados não estruturados disponíveis hoje em dia. A informação temporal está entre os diferentes tipos de informações que podem ser extraídos a partir de tais fontes de dados não estruturados, como documentos de texto. Informações temporais descrevem as mudanças que acontecem através da ocorrência de eventos, e fornecem uma maneira de gravar, ordenar e medir a duração de tais ocorrências. A impossibilidade de identificar e extrair informação temporal a partir de documentos textuais faz com que seja difícil entender como os eventos são organizados em ordem cronológica. Além disso, em muitas situações, o significado das expressões temporais é impreciso, e não pode ser descrito com precisão, o que leva a erros de interpretação. As soluções existentes proporcionam formas alternativas de representar expressões temporais imprecisas. Elas são, entretanto, específicas e difíceis de generalizar. Além disso, a análise de dados temporais pode ser particularmente ineficiente na presença de erros ortográficos. As abordagens existentes usam métodos de similaridade para procurar palavras válidas dentro de um texto. No entanto, elas não são suficientes para processos erros de ortografia de uma forma eficiente. Nesta tese é apresentada uma metodologia para analisar e normalizar das expressões temporais imprecisas, em que, após a coleta e pré-processamento de dados sobre a forma como as pessoas interpretam descrições vagas de tempo no texto, diferentes técnicas são comparadas a fim de criar e selecionar o modelo de normalização mais apropriada para diferentes tipos de expressões imprecisas. Também são comparados um sistema baseado em regras e uma abordagem de aprendizagem de máquina na tentativa de identificar expressões temporais em texto, e é analisado o processo de produção de padrões de anotação, identificando possíveis fontes de problemas, dando algumas recomendações para serem consideradas no futuro esforços de anotação manual. Finalmente, é proposto um mapa fonético e é avaliado como a codificação de informação fonética poderia ser usado a fim de auxiliar os métodos de busca de similaridade e melhorar a qualidade da informação extraída.

Palavras-chave: Extração de Informação, Normalização de Expressões Temporais Imprecisas, Similaridade Fonética.

Abstract

Information Extraction systems and techniques are able to deal with the increasing amount of unstructured data available nowadays. Time is amongst the different kinds of information that may be extracted from such unstructured data sources, including text documents. Time describes changes which happen through the occurrence of events, and provides a way to record, order, and measure the duration of such occurrences. The inability to identify and extract temporal information from text makes it difficult to understand how the events are organized in a chronological order. Moreover, in many situations, the meaning of temporal expressions is imprecise, and cannot be accurately described, leading to interpretation errors. Existing solutions provide alternative ways of representing imprecise temporal expressions, though they are specific and hard to generalise. Furthermore, the analysis of temporal data may be particularly inefficient in the presence of spelling errors. Existing approaches use string similarity methods to search for valid words within a text. However, they are not rich enough to processes misspellings in an efficient way. In this thesis, we present a methodology to analyse and normalise of imprecise temporal expressions, in which, after collecting and pre-processing data on how people interpret vague descriptions of time in text, we compare different techniques in order to create and select the most appropriate normalisation model for different kinds of imprecise expressions. We also compare how a rule-based system and a machine learning approach perform on trying to identify temporal expression from text, and we analyse the process of producing gold standards, identifying possible sources of issues, giving some recommendations to be considered in future manual annotation efforts. Finally, we propose a phonetic map and evaluate how encoding phonetic information could be used in order to assist similarity search methods and improve information extraction quality.

Keywords: Information Extraction, Imprecise Timex Normalisation, Phonetic Similarity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today there is a increasing amount of unstructured data being produced by different kinds of information systems, in a variety of formats, due to the advancement of communication and information technologies [Jellouli and Mohajir, 2011, Pavel and Euzenat, 2011]. Therefore, companies are becoming increasingly involved with the analysis of such volumes of data, which provides substantial competitive advantages for those who want to succeed in the era of a knowledge-based economy. Thus, science related to information management has evolved to develop new system modelling and building techniques for unstructured data formats, such as text documents [Wakil, 2002].

With the increased interest in finding and sorting information from text documents, text mining emerged as a technology with the purpose of extracting non-trivial knowledge from unstructured documents [Wakil, 2002]. Many text mining definitions can be found in different works, as (a) the study and practice of extracting information from text using the principles of computational linguistics; and (b) the process of extracting interesting knowledge, associations and non-trivial patterns from textual documents [Aranha, 2007].

Text analysis is one of the text mining techniques that has been widely used, which includes extracting information using algorithms to recognise the knowledge contained in the text, enabling software applications to use the content extracted from textual sources for various purposes. Information Extraction (IE) comprises the identification of specific information found in unstructured data sources, such as natural language text, and the classification and structuring into semantic classes, in order to make the information more suitable for information processing tasks [Moens, 2006]. The IE process starts by pre-processing text into a machine processable form, and then uses heuristics to identify the information to be extracted. IE can enable different applications such as question answering and information retrieval systems to offer more precise answers [Anantharangachar et al., 2013].

Information extraction from text involves multiple problems, including but not limited to: (a) how to identify strings representing the subjects in sentences, (b) how to disambiguate strings and assign them to the appropriate semantic classes, (c) how to extract the values for the various attributes from the text, and (d) how to connect events to a timeline, identifying temporal information concerning such instances.

Amongst different kinds of concepts that may be extracted, time is a primary element that allows us to observe, describe and reason about what surrounds us in the world, providing a substrate for the human management of perception and action [Caselli, 2009]. As a cognitive and linguistic component for describing changes which happen through the occurrence of events, processes, and actions, time provides a way to record, order, and measure the duration of such

occurrences. As a pervasive element of human life, the absence of a correct identification of the temporal ordering may result in a bad comprehension [Bartak et al., 2013, Caselli, 2009].

The extraction and understanding of temporal information from text is fundamental for language understanding [Burman et al., 2011] and an important sub-task for several language processing applications [UzZaman and Allen, 2010], such as text summarisation and knowledge base population. Processing a temporal expression (timex) from text, i.e. extracting and modelling the expression, includes tasks such as recognition and representation of the temporal information [Kolomiyets, 2012]. Solving challenging computational problems involving time has been a critical component in the development of information extraction systems [Bartak et al., 2013], e.g. understanding how such elements that describe temporal concepts can be formally represented and what procedures should be performed by an algorithm in order to deal with the set of operations that we as humans seem to perform relatively easy [Caselli, 2009].

Temporal information extraction from text is a challenging task, allowing to identify and to position extracted events in a chronological order. IE architectures should be able to identify expressions from the input text that represent temporality, making a connection between the events and a timeline. To understand how events are located on a timeline, it is necessary to find out two different things: (a) the events and (b) the temporal information conveyed in the text.

In many situations, however, temporal information cannot be accurately described, and imprecise data should be handled. Imprecise temporal expressions denote imprecise amounts of time, imprecise number of times or frequencies, and imprecise points in time (e.g. “less than one year", “many days", “about 3-4 times a week", “every 2 or 3 days", and “recently"). Although current guidelines for timex recognition describe rules to identify imprecise timexes in terms of language structure, the representation of imprecise temporal information in terms of value can be ambiguous or incomplete. Imprecise temporal expressions can represent up to 35% of the amount of temporal data in specific domains, such as in clinical narratives. The representation of imprecise temporal expressions in terms of value is the central problem in this research, and it will be described in the following subsection.

1.1 Problem Description

Extracting temporal expressions from text is a subtask in several different kinds of applications, such as extracting information from clinical narratives (medical records) and social media. In such medical record systems, clinical events are textually described and connected to temporal points in time. Recognising those clinical events and their temporal relations makes it possible to organise them in terms of a chronological order. However, in many situations, the meaning of temporal expressions is imprecise, and cannot be accurately described, leading to interpretation errors.

We illustrate this issue in Figure 1.1, which shows a sample of a medical record content (in Portuguese) extracted from the *InfoSaude*[1] system [Bona, 2002]. The example shows a set medical procedures (angioplasty and mammography), symptoms (insomnia) and drugs used (insulin), and it is a subset of sentences extracted from different patient's medical records in order to avoid patient's identification. Table 1.1 lists some of the extracted events from the previous medical record example in a chronological order, where T_0 indicates the current date or the documet creation time (DCT).

[1] *InfoSaude* is an information system created to manage and track medical records related to patient health, and it is used to manage over one million medical records in the public health system in Florianopolis, Brazil. Access to information stored in the *InfoSaude* system was granted according to the Confidentiality Agreement in Appendix A.

1:	HMP: Diabetes há cerca de 15 anos, tratamento com diamicron 2cp VO por dia. Sem uso de insulina.
2:	Paciente com IAM há 13 anos, realizou cirurgia de ponte de safena.
3:	Angioplastia com 6 stents há 6 anos, paciente relata que teve outro IAM durante procedimento.
4:	HAS há 30 anos trata com enalapril e selozok. Historia de pre eclampsia duas vezes.
5:	G6P4A2 partos normais e último {*cesarea*}. Fumante há 50 maços/ano. Nega etilismo.
6:	Paciente relata diversa internações por problemas {*cardiacos*}, não soube precisar a quantidade.
7:	Preventivo e mamografia ultima vez há dois anos.
8:	Pai falecido CA {*esofâgo*} 63 anos, irmão falecido sarcoma de coluna 23 anos.
9:	Paciente faz acompanhamento da Pa em casa, nega seguimento da glicemia em casa.
10:	{*Paceinte*} {*realata*} insônia e dificuldade no sono, acorda sem disposição.
11:	Nega angina de repouso. Uso de propoatilnitrato 3x ao dia.
12:	Monocordil 1cp de 12/12 horas. Dolamin toda noite devido a dor.
13:	Cilostazol 2x por dia após refeições. Trimetazidina 1cp por dia. Lorazepam 2mg 1 à noite.
14:	Clopidogrel 75mg uma vez ao dia. Omeprazol 20mg 11cp 12/12h. Enalapril 1cp 2x ao dia.
15:	Atorvastatina 1cp por dia. Selozok 2cp por dia.
16:	Relata perda de peso há cerca de 3 meses (relatou perda de 2kg).
17:	Há dois meses episódio de gripe, 2 dias de cama, uso de penicilina benzatin.

Figure 1.1: Medical record sample with temporal information and spelling errors.

Table 1.1: Sample of extrated events organised in a chronological order.

When	Line	Event
T_0 - 30 years	4:	hypertension
T_0 - about 15 years	1:	diabetes
T_0 - 13 years	2:	acute myocardial infarction
T_0 - 6 years	3:	angioplasty
T_0 - 2 years	7:	mammography
T_0 - about 3 months	16:	weight loss
T_0 - 2 months	17:	influenza

Organising events in a chronological order is important to find the temporal relations (e.g. before/after relations) amongst them.Temporal information extraction plays an important role in this respect. Temporal expressions (underlined expressions in Figure 1.1) are written in natural language and refer directly to time points or intervals – e.g. *"há 6 anos"* ("6 years ago") in line 3 –, serving also as anchors for linking concepts and events extracted from the text to a timeline, providing the correct distribution of such extracted elements in time [Ahn et al., 2005]. There are many situations, however, in which temporal information cannot be accurately described. In real life, there is uncertainty associated with the occurrence of many events. Thus, imprecise data should be handled. In line 16, *"há cerca de 3 meses"* ("about 3 months ago") is an example of imprecise temporal information found in the text.

Figure 1.2 ilustrates the importance of dealing with imprecise points in time. A query system performing searches over extracted events should be able to find those bounded by a certain period of time. Given two events e_1 and e_2, each one associated with a temporal expression t_1 and t_2, where t_1 is a precise DATE that makes it possible to place e_1 in a specific point within a timeline, and t_2 is an imprecise reference in the form "approximately N days later" which makes it impossible to know the exact day when event e_2 occurred. However, it can be reasoned the e_2 occurred after e_1. Considering a query that performs a search within the period bounded by q_b

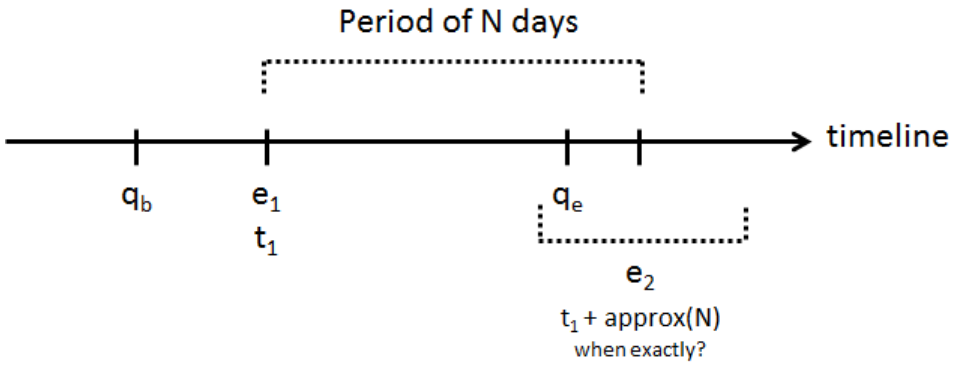


Figure 1.2: Example of an event (e_2) placed in an imprecise point in time.

and q_e, where: $q_b < t_1 < q_e$ and $q_e < t_1 + N$, we can surely affirm that e_1 would be part of the search result. On the other hand, it is impossible to evaluate whether or not e_2 has to be part of the same query result, as the numerical reference that surrounds the placement of e_2 within the timeline comprises a degree of vagueness that makes it impossible to say the exact date when e_2 happened.

Although some proposed approaches and systems can identify temporal information in text [Kolomiyets and Moens, 2013, Chambers, 2013, Bethard, 2013, Strötgen et al., 2013], they do not deal with imprecise temporal expressions, like “a few weeks ago" or “the coming months", in terms of defining more specific attributes to describe and connect those expressions to a timeline. Such approaches do not implement temporal-related logics to manipulate such inaccurate information, for example, to compare events associated respectively to expressions such as “about 2 months ago" and “a few weeks ago", indicating which one happened before or after [Ling and Weld, 2010].

Normalisation is the process of tagging a timex, setting attribute values that describe that expression in terms of an amount or a point in time [Kolomiyets and Moens, 2010]. Current annotation standards are restricted to normalise imprecise time expressions in terms of language structure or language elements [Ferro et al., 2005, Sauri et al., 2006, Pustejovsky et al., 2010, Styler et al., 2014]. An expression like “few weeks" is normalised to represent an “undetermined period of time" or an “undetermined number of weeks", making it hard to connect that expression to a timeline without any numerical value. When improving the normalisation guidelines to consider a timex description in terms of uncertain values or periods of time (e.g. range of values), events related to imprecise timexes can be chronologically placed, and temporal reasoning can be applied.

Moreover, misspelt words can be found in text. The textual content of medical records managed by *InfoSaude* does not go through any kind of review, leading to a number of spelling errors that could harm the analysis of the textual content, as “*cardiacos*" in line 6, or “*Paceinte*" and “*realata*" in line 10 (correct forms are “*cardíacos*", “*Paciente*" and “*relata*"). The IE process should deal with this problem to avoid loss of information. In Figure 1.1, words between { and } indicate spelling errors.

To overcome those spelling errors during the IE process, string similarity comparison algorithms can be used to identify concepts from the free text [Godbole et al., 2010]. Therefore, the problem of identifying temporal expressions in the text can still be more complicated when we consider that such expressions might also carry misspellings. Spelling errors of generic drug names, for example, can occur in up to one out of six entries in electronic drug information

systems, being responsible for up to 12% of adverse drug events, mainly caused by errors during transcription of prescriptions, illegible prescriptions, or drug name confusion. Due to such errors' frequency and the relevance of drug information in clinical tasks, automatic spelling correction and spelling error-tolerant engine systems can be useful to health care professionals [Senger et al., 2010]. Misspelt words can also affect the recognition of temporal expression within the text.

String similarity metrics can measure similarity between two text strings. Edit Distance (ED) [Levenshtein, 1966] and Jaro-Winkler (JW) distance [Winkler, 1990] are two well-known functions found in the literature that can be used to compare the elements from the input data source with an existing dictionary to identify a possible valid word for a misspelling. However, when working with unstructured texts, a complete inspection of a repository or dictionary during query execution is impractical in terms of processing time, and support to fast similarity search is also required, i.e. for a given possible not well-written word, we want to find similar words, but not performing a full search in the dictionary. Fast Similarity Search (FastSS) [Bocek et al., 2007] is an example of algorithm designed to find strings similarities in a database. It finds similar words from an input word with spelling errors based in the ED metric.

The existing string similarity algorithms coupled with a supporting dictionary may be inefficient, when the analysed text has spelling errors, because they may not necessarily handle specific aspects related to spelling errors, like phonetic errors. String similarity functions are not sufficient for time-sensitive applications, including enterprise and web scenarios, where typos, misspellings and noise must be processed in an efficient way [Stvilia, 2007]. In these cases, phonetic similarity metrics can be used in order to improve information quality. Phonetics are language-dependent [Ladefoged and Maddieson, 1996] and solutions for this sort of problems must be specially designed for each specific language.

This thesis addresses the overall problem of dealing with imprecise temporal expressions during the IE process. There is a lack in the current annotation standards regarding how to normalise imprecise timexes in terms of values. The normalisation of such time expressions would make it possible to use temporal-related logics and arithmetic to manipulate such inaccurate information. Thus, a normalisation methodology is necessary to describe how to capture the way people understand and reason about the vagueness carried by such kind of imprecise temporal data. Additionally, we also address the impact of spelling errors when trying to identify different concepts within the source text, including temporal expressions.

1.2 Objectives

The primary objective in this thesis is to present a methodology for the normalisation of imprecise temporal expressions. We collected data on how people interpret vague descriptions of time in text, and we compared different techniques in order to create and select the most appropriate normalisation model, which resulted in a grounded probability density function for the period over which the timex was attained.

As secondary objectives we also aim to:

- Evaluate distinct IE approaches: we compare how a rule-based system and a machine learning approaches perform on trying to identify temporal expression from text, depicting how a machine learning approach is able to reproduce manual annotations, including the mistakes made when manually creating the annotation standards.

- Analyse the process of producing gold standards: we identify possible sources of issues, and we give some recommendations to be considered in future manual annotation efforts, such as using a high recall rule-based system to automatically create annotations that can be represented by simple, unambiguous rules, to be further reviewed by human annotators.

- Evaluate how phonetics can improve similarity search methods: we extend a dictionary repository to support searching for phonetically similar words, and we analyse the impact of not dealing with misspelt words in the IE process, performing experiments to identify misspelt names of drugs and misspelt temporal concepts.

1.3 Contributions

The following contributions were achieved during the course of this research.

We describe two approaches we developed for time expression identification: a rule-based system that favoured recall and a machine learning approach built using readily available components, which was able to achieve a competitive F1 performance in a short development time. We discussed how they perform relative to each other, and how characteristics of the corpus affect outcomes and the suitability of the two approaches.

We suggest that inconsistent data in gold standards, such as those found in the Clinical TempEval corpus [Bethard et al., 2015], tend to lower the precision of rule-based systems. Thus, the appearance of a superior result by our machine learning system is therefore not to be taken at face value, as the machine learning system may have learned regularities in an incorrect annotation style, rather than having learned to accurately find time expressions. Machine learning systems have a flexibility and power in finding non-obvious cues to more subtle patterns, which makes them successful in linguistically complex tasks, but also gives them a deceptive appearance of success where the irregularity in a task comes not from its inherent complexity but from flaws in the dataset.

We examine temporal expressions in the Clinical TempEval, and the findings of this data-driven analysis are used as recommendations to be considered in future manual annotation efforts. We investigate where the annotation guidelines have proven difficult to apply, and give some recommendations regarding temporal annotation. We also detail the results of a principled analysis of expert manual annotations of temporal expressions in the THYME schema over a corpus of clinical notes, describing the main categories in which discrepancies between annotations and the guidelines were found.

The main contribution in this thesis is a normalisation methodology for imprecise temporal expressions extracted from text. Our methodology uses questionnaires to capture the way people understand and reason about the vagueness carried by imprecise temporal data. Answers are used as input data to produce histogram and fuzzy membership functions. Then, we compare statistical regression and machine learning techniques in order to evaluate which would be the most suitable model for each kind of imprecise temporal expression being evaluated.

We use F1-score to calculate how similar two membership functions are, and to choose the suitable representation model for each kind of imprecise temporal expression. We also propose a weighted F1-score variation ($F1_{3D}$) in order to identify where the differences are when comparing two membership functions. We apply the proposed methodology for three kinds of imprecise timexes, and we compare the resulted normalisation models between English and Portuguese. The way people understand that same kind of expression in two different languages can be different, and that discrepancy can be assigned to different aspects, including cultural

differences, the way the questionnaires are design, or different domains used to write the sentences in each questionnaire for each language.

We present an approach of fast phonetic similarity search coupled with an extended version of the Wordnet repository. Our approach has three main contributions: a) an indexed data structure (PhoneticMap), b) a novel string similarity algorithm (StringSim), and c) we integrated the previous contributions with Princeton WordNet (PWN) to implement the fast phonetic similarity search. *StringSim* is based on the notion of penalty, keeping the similarity values higher for words with less than 4–6 differences. In contrast, it decreases and converges the similarity values to zero faster comparing to other known string similarity metrics. We performe a set of extensive experiments in order to validate our approach. We apply our solution to a case study for discovering misspelt forms of drug names and timexes in a set of medical records. Fast phonetic similarity search has proven to be well adapted for combining string and phonetic similarity when finding misspelt words.

1.4 Organization

In Chapter 2, we describe IE systems, depicting the most common features and a general IE architecture, and comparing features and components used in some IE systems. Then, we describe the issues of identifying temporal information during the IE process, some of the existing solutions to identify and extract temporal information, and how some approaches were designed to identify and describe imprecise temporal data. Finally, we explore the problem of dealing with spelling errors in the IE process, evaluating an approach that uses phonetics as an attempt to improve similarity search methods, and how misspelt words interfere in the timex recognition process.

In Chapter 3, we present two approaches to time expression identification. The first is a comprehensive rule-based approach that favoured recall. The second is a machine learning system built using readily available components, which was able to achieve a competitive performance in a short development time. We discuss how the two approaches perform relative to each other, and how characteristics of the corpus affect the suitability of different approaches and their outcomes. Secondly, we examine temporal expressions in a clinical corpus. We investigate where annotation guidelines have proven difficult to apply, and give a series of recommendations regarding temporal annotation. Lastly, as the main contribution in this thesis, we propose a methodology for the normalisation of imprecise time expressions. After classifying the input data, we apply statistical regression and machine learning techniques that will produce a set of membership functions. The results are compared with an input dataset, producing an adapted score to guide the choice of the best model.

In Chapter 4, we introduce a novel approach for efficiently perform phonetic similarity search over large data sources. We use a data structure called *PhoneticMap* to encode language-specific phonetic information. This structure is used by a novel fast similarity search algorithm. We validate our approach through a set of extensive experiments. First, we executed different comparison scenarios to correct words with spelling errors, using a Portuguese variant of a well-known repository. Second, we applied our approach in a case study that identify misspelt drug names and misspelt time expressions in a set of medical records.

In Chapter 5, we conclude with the final considerations and future work.

Chapter 2

State of the Art

In Information Extraction (IE), relevant information from text is identified, collected and normalised. IE systems are built in order to analyse only those text parts that contain relevant information, recognizing entities and their associations. Such information is applicable to specific problems and situations. Thus, an exhaustive deep natural language analysis of all textual aspects is not always needed, it being possible even to skip irrelevant text passages [Maedche et al., 2002]. IE can use a diverse set of knowledge sources as part of the understanding process of a domain. The relevant information extracted from text can be used to design, enrich, and populate knowledge databases [Nedellec and Nazarenko, 2006]. Since most relations and events are temporally bounded, IE must also deal with temporal information [Mani et al., 2004]. IE systems may be able to extract temporal concepts and make the proper connections between relations and events and their position in the timeline [Moens, 2006]. The challenge in IE is to deal with imprecise temporal expressions, and being able to perform arithmetic and logical operations with such expressions. An IE system can also use dictionaries or lists of terms as a reference to identify temporal words and expressions from text. However, such repositories may not be sufficient to identify such terms whether the text has spelling errors. To support the process of identifying words and phrases with spelling errors, a good alternative is the use of semantic dictionaries accompanied by similarity comparison algorithms.

In Section 2.1, we describe IE systems, detailing the most common features in a general IE architecture. We also compare features and components of some IE systems to show how those components can be combined in different ways to produce specific solutions. In Section 2.2, we describe temporal concepts and the tasks that comprise temporal information extraction, highlighting the identification and annotation issues involved in such a process. We also show how fuzzy logic has been used to describe some imprecise temporal concepts. In Section 2.3, we explore the problem of dealing with spelling errors into the IE process, and how lexical databases can be integrated in the IE process to support similarity search.

2.1 Information Extraction Systems

Figure 2.1 shows an example of a commonly used architecture in information extraction systems. The raw text of a document is split into sentences using a sentence segmenter. Each sentence is subdivided into words (tokens) using a tokenizer and tagged with part-of-speech tags (POS tagging or POST). In the named entity detection step, the search for mentions of potentially interesting entities in each sentence is performed. Finally, relation detection is used to search for likely relations between different entities in the text [Bird et al., 2009].

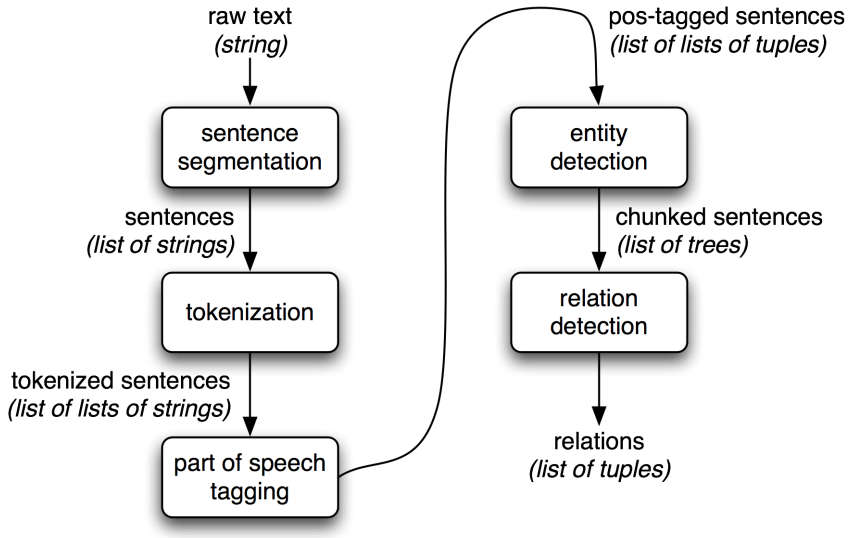


Figure 2.1: Example of a pipeline architecture used in IE systems [Bird et al., 2009].

2.1.1 Common IE Architecture

Implementation details of IE systems can be different from each other. Considering those differences usually found, it is possible to identify a common architecture of such systems [Wimalasuriya and Dou, 2010b]. Figure 2.2 shows the most common components we identified in IE systems. The common IE architecture is focused on textual input data processing, and it is designed based on the most common features and components, presented by other frameworks used in text IE applications.

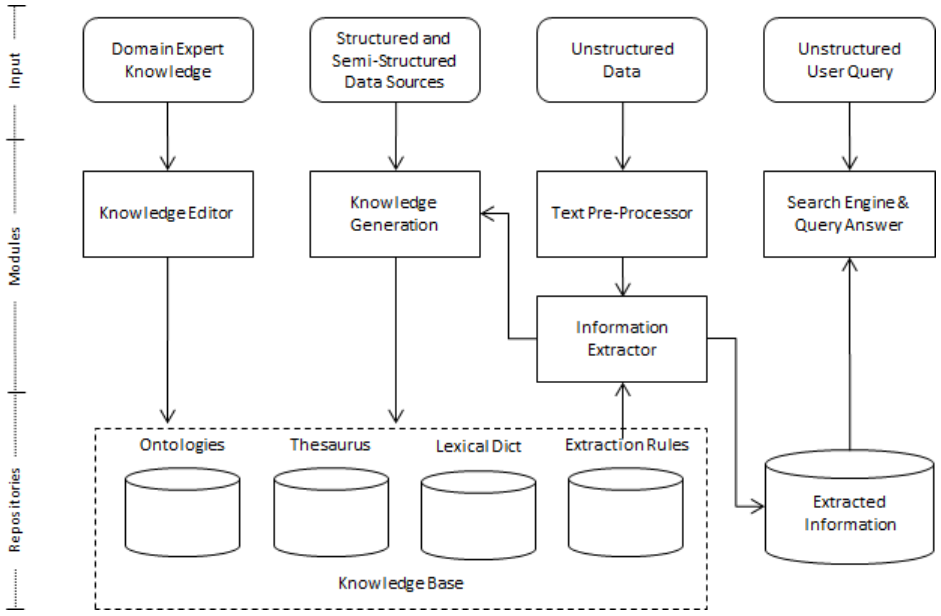


Figure 2.2: Common IE Components.

- The *Domain Expert Knowledge* refers to the knowledge (frequently informal and ill-structured) of a specialist or expert in a particular domain area, and must be transformed

into a computer readable form, as in a set of rules or interconnected concepts. The result of a computer activity can also be considered as expert knowledge when it refers to a specific domain and can be used to enrich a knowledge base [Hjorland and Albrechtsen, 1995].

- The *Knowledge Editor* module comprises all necessary tools to maintain the knowledge base, including ontology editors, dictionary search engines, and extraction rule constructors. Protégé [1] is an example of an ontology editor that provides a suite of tools to construct domain models and knowledge-based applications. Princeton WordNet (PWN) is a commonly used digital lexical dictionary for the English language created to produce a more intuitively usable combination of dictionary and thesaurus, and to support automatic text analysis [Miller, 1995]. JAPA (Java Annotation Patterns Engine) is a version of CPSL – Common Pattern Specification Language – that provides finite state transduction over annotations based on regular expressions, and can be used to design and develop extraction rules.

- The *Knowledge Base* (KB) is an information repository that provides a means for information to be collected, managing all the knowledge sources and rules used in the IE process, usefully comprising one or more of the following repositories: ontologies, thesauri, lexical dictionaries, and extraction rules.

    1. The *Ontology Repository* provides ontology content-based information about the knowledge domains involved in IE. Ontologies provide a conceptual representation of knowledge within a given domain, comprising a controlled vocabulary of concepts, their properties, relationships and restrictions, which can be used to describe and carry out inferential reasoning about the domain [Gooch, 2012]. An Ontology is a hierarchical definition of objects (classes), properties and relations between them, representing the structure and formal specifications of the concepts in a specific knowledge domain. In other words, it is an “explicit specification of a conceptualisation" [Gruber, 1993].
    2. The *Thesaurus Repository* defines a vocabulary for different kinds of terms and relationships, basically mapping words to instances and listing words that are indicative of relationships (properties). Unlike a dictionary, a thesaurus does not give the definition of words, and it just lists words according to a similarity of meaning or subject.
    3. The *Lexical Dictionary Repository* can store semantic dictionaries to represent interrelated words and concepts in different languages, as the example of using WordNet [Gomes et al., 2004] – Section 2.3.2 gives a more detailed description of WordNet.
    4. *Extraction Rules* can be represented as regular expressions or specific algorithms for atomic and complex data type information extraction, including dates, times, integer and float numbers, IP numbers, proper names, possible instance relationships and others, i.e. everything the thesaurus structure itself cannot represent [Wimalasuriya and Dou, 2010a]. An extraction rule works as the metadata for those concepts, without the need for a training set.

- *Structured and Semi-Structured Data Sources* may contain different types of data structures as structured databases or semi-structured XML documents [Ghawi and Cullot, 2007].

[1] `http://protege.stanford.edu/`

Structured data and metadata from information systems, and semi-structured documents can be used to generate content of the KB. Conceptual entities, relationships, attributes and a subset of tuples represent a significant contribution to the description of specific knowledge domains managed by IE systems, serving as the main input of the *Knowledge Generation* module.

- In the *Knowledge Generation* module, structured databases, semi-structured documents, and also information extracted from text during the IE process are mapped to populate the KB repositories and express the semantics of such information sources.

- IE uses *Unstructured Data* as its main input. Text-based information can be acquired from the Internet (web pages) or information system databases which store unstructured textual fields (e.g. medical record management systems), being used as a main information source of the information extraction process itself.

- The textual input usually goes through a pre-processor component that converts the text to a format that can be handled by the IE module. The *Text Pre-processor* is usually used to perform those tasks that include removing specific tags, splitting text into sentences and words, and text annotation as POS tagging. As an example of text processing tool, GATE (General Architecture for Text Engineering) [Cunningham et al., 2011] is a framework for the development of large scale natural language processing pipelines. It processes documents in different formats (plain text, HTML, XML, RTF, . . .), and it also has a widely used NLP toolkit that can be used to directly deploy extraction rules [Li et al., 2009].

    1. When unstructured text data come from internet or any other semi structured format (as HTML or XML), a specific cleaner must be applied to text input to treat or remove specific tags, or remove parts of the text that may be considered as not relevant for the analysis.
    2. When splitting the text, long sentences can be decomposed into smaller clauses or words, distinguishing relevant clauses from non-relevant ones, identifying less salient entities in subordinate clauses, and helping to interpret the meaning of entities in a text based on the knowledge of textual structure. This makes it possible to connect entities and words from different clauses, improving the performance on long-distance dependency paths [Maslennikov and Chua, 2007].
    3. Text annotation is the practice of adding tags to a text. Text is tagged based on the basic concepts available in the thesaurus repository, linking the vocabulary with corresponding classes and properties definitions from an ontology, and word-based and expression-based mapping, like identifying integers and float numbers, date and times, and other forms of basic information structures by using extraction rules. Moreover, texts can be pre-processed using specialised algorithms, such as stemming and stop word removal algorithms, in order to save computing time [Latha et al., 2007].
    4. In the context of text annotation, POS tagging, also called word-category disambiguation, is the process of marking up a word in a text as corresponding to a particular part of speech. This task is based on both definition and context. Since a large percentage of word-forms are ambiguous, POST is harder than just having a list of words and their parts of speech. Many words can represent more than one part of speech at different times. POST can be used to differentiate word senses that involve

part of speech differences, and in pre-processing text to speed up parsing [McCallum, 2006].

- The *Information Extractor* module uses annotated text and extraction rules to find conceptual instances, their properties and possible relations based on the tags included in the source text [Ashish et al., 2009]. Templates are used to define information of interest and can be also used in conjunction with common search algorithms to match text pieces, and heuristics that make this search feasible [Corney et al., 2008]. Once named entities were identified in a text, it is possible to extract the relations that exist between them, and regular expressions can be used to identify those instances that represent specific relations [Bird et al., 2009].

- The *Extracted Information* is a repository that stores the result of unstructured data analysis. The text-based information extraction result must be represented in a structured form. Such structured data representation must consider all possible valid combinations of extracted events and their relationships from the input text. New and previously unknown information can be found and identified from unstructured texts. New information can be used to populate the KB with new instances [Yankova et al., 2008].

- The same way that IE handles unstructured textual data as input source, user queries can also be written in an unstructured textual format. An *Unstructured User Query* can be treated in the same way as text input and its result can be used as a parameter to search over the repository of extracted information. Unlike [Wimalasuriya and Dou, 2010b], that considers *Unstructured Query Answer* as an external part, we consider it as part of an IE system, largely interconnected to the *Search Engine & Query Answer* component.

- Lastly, IE systems are usually a part of a larger query-answering system, in which the output is often stored in a database or a knowledge base. The query answering system makes use of the extracted information, stored either in a knowledge base or a database, and a reasoning component to answers user queries [Wimalasuriya and Dou, 2010b]. The *Search Engine & Query Answer* component is responsible for transforming user queries from unstructured to a structured format and for matching concepts to the repository of extracted information, retrieving not only exact matches, but also structurally similar concepts [Lucrédio et al., 2012]. Using the *Information Extractor* component to promote the conversion of a query to a structured model, the query can be compared with data stored in that repository to generate a list of possible results.

2.1.2 Examples of IE Systems

Comparing features and components used in different IE systems, one can observe that some required components, such as Knowledge Base and Text Pre-processing, are found in all of the solutions. Some of the non-mandatory components are used as needed by each system to perform specific operations during the extraction process, and others are applied according to the purpose of the system, such as the extraction of temporal information. The KB is a required component comprising the arrangement of different repositories. Text pre-processing features vary according to the purpose of each system, and may also include further processing tasks, such as a gazetteer (matches proper nouns), and co-referencer (finds identity relations between entities, e.g. “he", “it", “them").

MUSING [Saggion et al., 2007] is a project founded on semantic-based knowledge and content systems, which integrates Semantic Web and human language technologies for enhancing

knowledge acquisition and reasoning in Business Intelligence (BI) applications. It uses time and domain ontologies for guiding information extraction onto an ontology population task, acting over the knowledge repository for generating new knowledge. The integrated reasoning architecture considers temporal aspects, which in are MUSING a central point of interest. In MUSING, documents are processed by an Ontology-based annotation tool to detect information about companies, countries, and regions.

OntoText [Anantharangachar et al., 2013] is a system that uses a knowledge extraction procedure to extract ontology instances from a set of the text documents. It uses the concept of a Semantic Lexicon to identify the semantic domain. A lexicon is a set of words and it usually is not specific to any domain. A semantic lexicon is basically a set of words that are domain specific and can identify a domain uniquely. After the domain is identified, the instance extractor module extracts the instance information to populate an ontology. The Lexicon learning/extractor module learns new lexicon symbols from the text, using a set of heuristics to identify lexical items that are related to the existing semantic lexicon. Using simple to complex pattern matching techniques to extract name value pairs, OntoText creates new instances and assign them the various values that were extracted from the text.

TextPresso [Muller et al., 2004] is a text-mining system for biological literature. It splits a collection of the full text of scientific articles into individual sentences, and allows searching over a database of articles and individual sentences based on categories of terms. A search engine enables the user to formulate semantic queries to search for one or a combination of these keywords within a sentence or document. TextPresso identifies terms that are stored in a lexicon according to a set of predefined categories and marks up the terms by enclosing them in XML brackets. The TextPresso search engine allows the user to select a combination of categories, subcategories, and keywords and submit queries to the whole publication or to a sentence. It was tested with the automatic categorization of papers according to the types of biological data they contain, and retrieving sentences containing a specific type of biological data from text.

TRIPS [Blaylock et al., 2011] is a statistical-symbolic natural language (NL) parser that uses Semantic Web technologies in an IE framework to automatically extract a timeline of a patient's healthcare events from a set of clinical notes. The system includes a legacy, generic, linguistically oriented ontology, a logical form output, and graph-matching extraction rules to Semantic Web technologies and representations. A semantic representation makes the framework more robust, allowing better reasoning, and the authoring of more general rules for extracting information of interest from the parsed text. Extraction rules are then applied to match and extract instances from the KB. These instances are stored in a second KB which allows a compact representation of the extracted information as well as further reasoning on the information.

RExplore is a tool for exploring information about research, which provides novel visualisation and navigation mechanisms to facilitate an understanding of the dynamics of research areas (e.g. main trends, new emerging topics, significant research 'shifts', authors ranking, etc.). From a research perspective, RExplore aims to provide a platform for testing some new visual analytics and knowledge-based methods, comparing them to other existing solutions. From a user point of view, it aims to provide a new tool to allow people to explore research data. RExplore makes it possible to rank Semantic Web authors by number of publications, to visualise a variety of relations between authors, and to understand the evolution of a topic over time [Motta and Osborne, 2012]. In RExplore, a resulting knowledge base is generated using statistical methods and background knowledge based on a large-scale corpus of publications, and after augmenting with geographic information. It also introduces a fine-grained, automatically populated topic ontology, in which topics are identified and structured according to a number of semantic relationships [Osborne et al., 2013].

The Drug IE System [Li and Shen, 2009] extracts specific information from the body of drug-related texts according to the relations, concepts and keys defined in the input ontology, usually built by a domain expert. The Parser module parses the domain knowledge to identify source information stored in a database. Information extraction rules are generated automatically according to the results of the Parsing and Dictionary Editing modules. The system does pre-treatment to the input files, performing lexical and syntax analysis and simplifying the input file for the information extraction step, which stores the results into database. After that, the system can inquire and make statistical analysis of the results.

Figure 2.3 summarizes features and components found in each analysed system.

Features		OBIE Systems and Architectures					
		Musing	OntoText	Textpresso	TRIPS	Rexplorer	Drug IE System
Data Formats	XML/OWL	X	X	X	X	X	X
Tools	Protege		X				
	GATE	X					
Ontology	Population	X	X			X	
	Auto Selection		X				
Text Preprocessor		X	X	X	X	X	X
	NLP				X		
	Sentence Split			X			
	Text Annotation	X		X			
	POS Tagging				X		
Knowledge Base		X	X	X	X	X	X
	Semantic Lexicon		X	X	X		
	Wordnet				X		
	Extraction Rules		X		X	X	X
Temporal Information*		X			X	X	
Spelling Errors		No	No	No	No	No	No

* No mention about imprecise temporal data

Figure 2.3: IE common features and components.

Some systems address the issue of extracting temporal information. Although imprecise time expressions can be identified in text during the IE process, current approaches do not completely deal with imprecise temporal expressions. Furthermore, none of the analysed solutions explicitly consider manipulating texts with spelling errors.

2.2 Temporal Information in the IE Process

Time provides a substrate for the human management of perception and action. As a pervasive element of human life, time is a primary element that allows us to observe, describe and reason about what surrounds us in the world [Caselli, 2009]. As a cognitive and linguistic component for describing changes which happen through the occurrence of events, processes, and actions, time provides a way to record, order, and measure the duration of such occurrences [Bartak et al., 2013].

Understanding temporal information is a subtask in language processing applications, such as question answering, text summarisation, information retrieval, and knowledge base population. To this end, it is important to develop strong annotation standards and corpora for temporal semantics. Challenges in developing these standards include: a) how to formally represent the elements that describe temporal concepts, and b) what procedures should be performed by an algorithm, in order to deal with the set of temporal reasoning operations that humans seem to perform relatively easily [Caselli, 2009]. The sub-problem of automatic recognition of temporal expressions within natural language text is a particularly challenging and active area in computational linguistics [Pustejovsky et al., 2003a].

The general process of reading and understanding a text includes inference about whether the presented situations, standing in a particular temporal ordering, precede, overlap or are included one within the other. Nevertheless, this seemingly takes into account a set of complex information involving different linguistic entities and sources of knowledge, such as [Caselli, 2009]:

- relevant situations that can be recognised as being involved in a temporal relation and those which are not;
- different entities with different ontological status that can be related, like things that happen in the world, and temporal expressions;
- relevant information that can determine the actual temporal relation.

Processing temporal information from text comprises extracting and representing such expressions in order for it to be used by a machine, regarding concepts of temporal cognition such as time, events, and relations between events and times. Temporal information in text is implicit, i.e. textual documents do not provide temporal information directly, as one can find in structured data sources such as timestamps. To identify temporal information in text we deal with the tasks of identifying, recognizing and representing such information [Kolomiyets, 2012]. According to [Sun et al., 2013], temporal reasoning is a non-trivial and usually complex task, mainly due to:

1. Temporal representation: Finding a design that can describe all possible temporal information, supports temporal inference, and computes efficiently is not easy. To design a time ontology, one needs to consider the many options for modelling the structure of time (linear, branching, or circular), to determine whether time has start/terminal points, to choose between continuous or discrete representations of time, and to decide whether to use time points or intervals as temporal references.
2. Complexity of temporal representation in natural language: Temporal information needs the integration of multiple levels of linguistic processing (grammar, semantics, discourse, and inference) to be understood, as it is expressed in language by a variety of means, and requires general conceptual knowledge. Some problematic aspects about the way temporal information is represented in natural language, such as: a) underspecified

temporal relations, vagueness of tense and aspect, relative times, implicit event durations, and temporal aggregates. Natural language analysis is difficult due to some domain specific aspects, e.g clinical narratives, as being ungrammatical, and having lots of abbreviations and misspellings.

Recent emergence of language processing applications like question answering, information extraction, and document summarization has been drawing attention to systems that are temporally aware. Extracting temporal information from raw text is fundamental for deep language understanding [UzZaman and Allen, 2010]. A retrieval system that is more aware of the temporal information allows users to get the results based on different time contexts, instead of using simple creation or last modified date attributes. Applications that can benefit from using temporal information in different ways range from ad-hoc retrieval to exploratory search [Alonso et al., 2007].

Answering questions carrying temporal aspects, which must be considered as query parameters, requires the extraction of temporal information encoded in natural language text entered by a user. Not only explicit temporal information must be extracted, such as dates and time expressions, but also implicit temporal expressions have to be correctly annotated (e.g. the time reference in "which were the most used drugs *last year*?"). A temporal question answering system also needs to have access to the temporal relations between events, to deal not only with *when* questions, but also relative temporal questions such as "what happened before <event>?" [Schilder and Habel, 2003].

In natural language, temporal information is not always stated explicitly. It is often implicit, hidden in the text, requiring interpretations or inferences using world knowledge and assumptions. Handling implicit time and related issues is an intermediate step, usually following the extraction of explicit temporal assertions about events, and prior to any logical or mathematical reasoning mechanism. A major challenge is to extract and combine events with the temporal information supplied by a temporal tagger [Zhou et al., 2006].

Temporality is a contextual information that plays a critical role when extracting information from narrative text documents [Meystre et al., 2008]. As an essential dimension for the interpretation of clinical narratives, time provides a context that makes meaningful the order in which the symptoms develop, the timing of different treatments, and the duration and frequencies of medications [Sun et al., 2013]. Temporal modeling and reasoning has been the focus of many recent studies, showing semantic models of temporality are needed to understand disease progression, adverse drug reactions, and other clinically relevant events over time [Velupillai et al., 2015].

2.2.1 Temporal Information Extraction

Temporal information extraction is a subtask of IE, in order to extract time expressions and temporal relations from natural language text and represent them using certain knowledge frameworks, normalising implicit expressions in text according to the embedded temporal relations, and assigning the appropriate temporal attributes to higher level information entities such as events.

A temporal expression (timex) is a sequence of tokens (words, numbers and symbols) that represents an instance of a temporal entity, which describes a point in time, duration or frequency [Sanampudi and Kumari, 2010]. A timex denotes *when* something happens, *how often* it happens or *how long* it lasts [Kolomiyets, 2012]. As a linguistic expression, a timex refers to a point in time, period, or recurring pattern in time [Llorens et al., 2012], as described on Table 2.1.

Table 2.1: Timex taxonomy [Llorens et al., 2012].

Timex type	Description
Explicit, absolute, or self-contained	Timexes can be directly translated to a particular date/time granularity
Implicit, relative, or context-dependent	Timexes need the document creation time (DCT) or other temporal reference/anchoring to obtain an explicit date/time
Durative	Timexes describe a bounded interval (duration) that is not inherently anchored to a timeline
Set or frequency	Timexes refer to regularly recurring times, such as “every Christmas" or “each Monday"
Vague	Timexes represent generic mentions like “recently" or “nowadays" and are usually annotated as a past, present or future date/time reference

Timex processing consists of recognising temporal expressions in text and their interpretation, which results in an annotation encoding a standardised representation of the timex semantics [Llorens et al., 2012]. Extracting temporal information from text is not a single task, comprising a number of sub-tasks that can be defined in different ways depending on the context in which it is applied.

Temporal information processing is a requirement for temporal question and answering (e.g. “when...?" and “how long...?"), and ordering events chronologically on a timeline. Additionally, [Wong et al., 2005] highlights three major challenges in temporal information extraction research:

- Difficulties in linguistics: there are many ways to express time, and not all times are exact, e.g. “two weeks ago" could represent (a) the entire week two weeks before the current one, (b) the day two weeks before the current one, or (c) a day approximately two weeks before the current one.
- Reference resolution: reference can be categorized into (a) relative time (e.g. “two days before last Christmas"), (b) explicit anchoring event (e.g. “two days before we met"), and (c) reference without an explicit anchor (“two days ago") that refers to a global reference time, which may be based on event time, speech time, or publishing time.
- Negation processing: considered a problematic language phenomenon, negation is ambiguous in language understanding mainly because (a) negation scope often overlaps with the scope of the quantifiers and tense operators, making it hard to determine which of the negated phrase is in the sentence; and (b) a semantic analyser has to interpret the negation, which can includes negation of event, object, quality, situation, location, manner, context, etc.

According to [Fagerberg, 2014], the temporal information extraction process comprises: a) temporal expressions have to be recognized within some kind of document and extracted from it; and b) extracted temporal expressions should be categorised and normalised to a canonical form – normalisation is not just a formatting problem, but a task in which the appropriate value of the extracted expression has to be calculated.

In [Kolomiyets, 2012], temporal information processing is divided into four sub-tasks: a) recognition of temporal expressions that denote time; b) normalisation of temporal expressions, in which a calendar value can be estimated; c) recognition of events that can be organized chronologically; and d) recognition of temporal relations that link events and times (event-time link) or organize events according to their temporal order (event-event link).

In [Sun et al., 2013], a NLP-based temporal reasoning is defined as a combination of: a) a temporal representation formalism – a machine-readable representation of the temporal dimension that includes the notion of time, temporal events, and possible temporal relations; b) the extraction of temporal information from natural language text – automated annotation and normalisation of temporal information from natural language texts based on a formalised representation; and c) a temporal inference over the extracted information – logical deductions performed on the extracted temporal information to enhance natural language understanding.

Timex Recognition

Definition 2.1 (Timex Recognition) *Timex Recognition (or Timex Annotation) is the task of finding the corresponding labels* $(y_1, ..., y_n)$ *to a given input string of tokens* $(x_1, ..., x_n)$ *so that the resulting labelling can be decoded into textual spans that constitute the tokens and denote time in the input string [Kolomiyets, 2012].*

Regarding to timex recognition, different methods, such as rule-based or machine learning techniques, can be employed in order to identify the spans of phrases in text that relate to time [Kolomiyets, 2012].

According to [Alonso et al., 2007], three primary temporal expression categories can be identified: (a) explicit, that includes entries in some timeline, such as an exact date or year (e.g. “January 2013"), (b) implicit, which encompasses a temporal information reference that can be anchored in a timeline, such as names of holidays (e.g. “Christmas 2012"), and (c) relative or indexed, when entities can only be anchored in a timeline in reference to another explicit or implicit one (e.g. “today", “now", “next month", “three days ago"). [Schilder and Habel, 2003] also considers a fourth classification, when temporal expressions express only vague (or imprecise) temporal information and it is hard to place the expressed information on a time line (e.g. “several weeks", “in the evening"). The challenge is to provide a model in which temporal expressions can be represented in an expressive, sound and unambiguous way [Zhou et al., 2005].

Temporal annotation is a complex task that makes high-quality temporal annotation impracticable or unrealistic when it relies only on human annotators. Instead, one can combine the strengths of human and machine to cooperate in a mixed-initiative annotation effort [Verhagen, 2004].

Based on an exhaustive analysis of 147 clinical records, [Zhou et al., 2011] summarizes the common types of temporal expressions, establishing temporal expression classification from such expressions. Simple temporal expressions are divided in three categories, which can also be combined to form composite time. Despite including uncertain temporal expressions in the following classification, the authors state that the automatic extraction work was hampered by the existence of such expression type. Table 2.2 describe categories of simple temporal expressions, and Table 2.3 shows composite temporal expressions found.

Other researchers have annotated temporal information in clinical text. For example, the CLEF Project [Roberts et al., 2009] semantically annotated a corpus to assist in the extraction of clinical information from text. It used two different schemas to annotate a) clinical entities and relations between them, and b) time expressions and their temporal relations with the clinical entities in the text.

Table 2.2: Categories of simple temporal expressions [Zhou et al., 2011].

Class	Sub-class	Freq	%	Example
Specific time expression	Date	416	30.17%	1992-7-6, May 3,1988, August
	Time of day(TOD)	98	7.11%	21:00, 17:00
	Duration(Dur)	233	16.9%	3 months, 4 days
	Age	6	0.44%	at 3 years old, at the age of 47
	Duration Range(DurR)	19	1.38%	2-4 days, 2-3 weeks
	Date Range(DateR)	2	0.15%	the year 1987 to 1998
Fuzzy time expression	Duration as Time Point (DurTP)	99	7.18%	2 weeks ago, after 4 days, 2 years ago
	Relative Time(RTime)	183	13.10%	yesterday, today, last year, this month
	Past, Now, Future(PNF)	30	2.15%	at present, recently, in the past, in recent days
	Part of day(POD)	112	8.12%	morning, afternoon, at night
	Unspecified Duration(UDur)	2	0.15%	several months, several decades
	Season	19	1.38%	spring, winter, autumn and winter
	Modified Date(MDate)	23	1.67%	early in 1980, middle of March, late in October
	Modified Duration(MDur)	82	5.95%	almost 5 months, more than 2 months
(EBT)	Event-based Time	55	3.99%	3 days before admission, 6 months after surgery

Table 2.3: Composite time statistics [Zhou et al., 2011].

Composite	Frequency	Percentage	Example
Date+POD	11	5.42%	morning, March 12,2001
Date+POD+TOD	5	2.46%	11:00am, April 2
Date+Season	9	4.43%	autumn,1998
Date+TOD	52	25.62%	14:20, July 3
POD+TOD	18	8.87%	8:00pm
RTime+Date	37	18.23%	last July14
RTime+Date+POD	4	1.97%	last month, 11 am
RTime+POD	45	22.17%	last afternoon
RTime+Season	6	2.96%	last summer
RTime+POD+TOD	3	1.48%	2:00pm,the day before yesterday
RTime+MDate	13	6.40%	late October of this year

Timex Normalisation

The normalisation task consists of obtaining the absolute value of a timex regardless of the linguistic expression used [Llorens et al., 2012]. After a TIMEX is recognized, its temporal value must be defined, which means finding the TIMEX3 value attribute for such temporal expression. The normalisation process is usually implemented as a rule-based system to overcome some problems, including: a) the infinite number of possible labels, and b) the large number of ways a calendar value can be expressed in natural language [Kolomiyets, 2012].

Rule-based approaches are traditionally used in timex normalisation. [Kolomiyets, 2012] presents a normalisation technique that comprises three sub-tasks:

1. Timex type classification, discourse type and token labelling: a classifier has to distinguish between 4 different labels of DATE, TIME, DURATION and SET, to define the type of time expression; it also uses a rule-based method to perform the semantic analysis of time expression constituents (token labelling), identifying different categories (Table 2.4) with a comprehensive vocabulary and a set of context dependent normalisation rules specific for that category.

2. Estimation of temporal values: temporal values are estimated (normalised). This is not considered a difficult task for absolute temporal expressions, because such kinds of timexes contain all components required for calculating the final value. Relative expressions (“last week", “next month") also can be represented using ISO standards [ISO, 2007] representation facilities.

3. Aggregation of temporal values: an aggregation of temporal values is performed, when one temporal expression consists of a set of shorter temporal expressions that are obtained by pre-normalisation; in this case, partially estimated values are aggregated to obtain a final temporal value.

Table 2.4: Timex categories [Kolomiyets, 2012].

Category	Examples
Temporal units	day, month, year
Temporal modifiers	last, previous, next
Temporal quantifiers	several, few
Temporal directions	ago, further, later
Temporal approximators	almost, about
Day names	Monday, Tuesday
Month names	January, February
Cardinal numbers	one, 1, two, 2
Ordinal numbers	first, 1st, second, 2nd
Coreference timex	period, time
Fixed timex	today, yesterday, now

Although it is easy to recognize such expressions with supervised machine learning, normalisation (interpreting them accurately) is a complex task that requires human knowledge, since any practical approach to timex normalisation requires a hand-crafted rule set [Llorens et al., 2012].

Temporal Relations

Temporal Relation Recognition or Temporal Linking is the task of finding temporal relations between temporal elements, such as events and times in a document. Temporal relations can occur in four possible entity pairings: event-event, event-time, time-time, and event-DCT (document creation time) [Chambers, 2013].

The set of temporal relations in text originates in the work of Allen [Allen, 1983], who defined thirteen temporal relations based on how an ordered pair of intervals can be associated [Kolomiyets, 2012].

Allen’s Interval Algebra [Allen, 1983] is an algebraic framework for qualitative reasoning with time intervals and expressions, in which temporal aspects of events in natural language text can be represented using a subset of 13 basic relations, which impose ordering constraints on the intervals: before(p), meet(m), overlap (o), starts(s), finishes(f), during(d), equal(e), during-by(D), Overlapped-by(O), Started-by(S), Finished-by(F), Met-by(M), after(P). Allen’s interval checking procedure within backtrack search have been shown to be limited in the sense that it can only represent relative ordering of the intervals but not its duration and negation of temporal information such as “patient will *not* use this medication *for the next two days*".

Allen's temporal relations between intervals are depicted in Figure 2.4, which shows seven relations and the inverses of six of them. An interval X can also be represented as a pair of points (x_1, x_2) where x_1 is the begin point, x_2 is the end point and $x_1 < x_2$, making it possible to rewrite basic relations using precedence and equality relations on begin and end points [Verhagen, 2004].

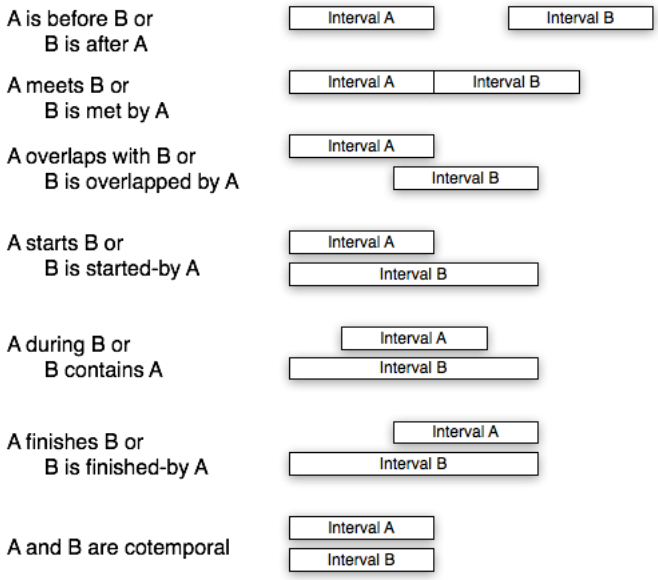


Figure 2.4: Allen's temporal relations [Allen, 1983].

Temporal entities can be interpreted as "special" measures of time to associate what happens in time and in the world. Two main subcategories of temporal entities are identified: interval and instants, both considered as ontological primitives. Intervals have durations but instants do not have an internal structure, with respect to intervals. Temporal relations are a direct consequence of time, by means of the relation of precedence, and can be identified between the temporal entities: a) relations between intervals; b) relations between instants; and c) relations between instants and intervals [Caselli, 2009].

Relations between intervals can be described and defined following Allen's standard interval calculus. Allen [Allen, 1983] argues that points are not necessary since all events can be decomposed, even for those events that appear to be a precise point in time, making it perfectly alright to use short intervals in the logic.

Instant relations can be defined in the exactly same way as for intervals, with the difference that they reflect the idea that "instantaneous" moments of time that their beginning and end points are the instants themselves. For instant relations, there is a reduced set of relations: *equal* or *simultaneous*, *meets/is met by* and *before/after*. The restricted set of instant relations prevents to contradict instants and intervals are ontological primitives. Otherwise, instants would be conceived and derived from intervals if all the other relations existed [Caselli, 2009].

Temporal information and relations between events that are vague or unclear make it more difficult to develop a chronological representation, and point some problems in attempting to place these events on a timeline. Very often, events are described without precisely defined time boundaries, i.e. exact start and end dates are unknown. When not all events can be associated with a date, placing events on a timeline can easily lead to false interpretations about their temporal relationships [Bassara, 2007].

2.2.2 Temporal Annotation Guidelines

The development of temporal annotation standards and corpora has a long history. Of note is the TimeBank corpus [Pustejovsky et al., 2003c], which contains 183 news articles annotated with temporal information, events, times and temporal links between events and times. This corpus was developed in multiple iterations, and prior analyses of the annotated data and the annotation standard aided the evolution of both. For example, [Boguraev and Ando, 2007] presented an extensive analysis of the TimeBank reference corpus in terms of development support of TimeML-compliant analytics, which helped advance the state of the art in temporal annotation. Indeed, iterative application of an annotation standard and examination of the resulting annotated data are critical steps in the MATTER development cycle, used for construction of annotation standards [Pustejovsky, 2006, Pustejovsky and Stubbs, 2012].

ISO SemAF

ISO SemAF (Semantic Annotation framework) 8601 [ISO, 2007] is an international standard for representing dates and times using a combined representation in the format YYYY-MM-DD'T'hh:mm:ss. Durations can also be represented using the format PnYnMnDTnHnMnS (or PnW) – P is a designator for the period, n is the value for each date and time element, and Y, M, D, W, H, M, S are the duration designators for number of years, months, days, weeks, etc. Parts of day, weekend, seasons, decades and centuries were introduced as new concepts in the calendar.

The extended version of ISO 8601 also provides underspecified or unknown values, by using a placeholder character X for those calendar field values when the context does not allow the values to be specified (e.g. 2012-01-XX for January, 2012), and values for temporal expressions that refer to the past, the present or the future (e.g. "nowadays", "lately", "recently") defined by alphabetical tokens PAST_REF, PRESENT_REF and FUTURE_REF for referencing to the past, the present and the future respectively [Kolomiyets, 2012].

STAG

STAG (Sheffield Temporal Annotation Guidelines) [Verhagen, 2004] was proposed as a means to annotate events, time expressions and the relations between them, classifying events in four groups: occurrences, perception events, reporting events and aspectuals. All events and time expressions are related with one of these three tags: relatedToEvent, relatedToTime, and relType. For the first two, values reference other events or time expressions. The last one contains a restricted value to identify the relation type (BEFORE, AFTER, INCLUDES, IS_INCLUDED or SIMULTANEOUS). This last relation type is intended to be fuzzy and include all kinds of overlaps.

Example: *The plane crashed on Wednesday*

```
The plane
<event eid=9 class=OCCURRENCE tense=PAST relatedToTime=5
relType=IS INCLUDED>crashed</event>
<timex tid=5>Wednesday</timex>
```

TIDES

TIDES (Translingual Information Detection, Extraction, and Summarization) is defined as a set of annotation guidelines for time expressions with a canonicalised representation of the

times they refer to [Verhagen, 2004]. TIDES is a standard that specifies kinds of markable and not markable expressions, and how to capture the semantics of temporal expressions. TIDES introduced the <TIMEX2> tag and a set of tag attributes to identify temporal expressions in text and their related information [Ferro et al., 2005]. Moreover, TIDES standard also describes how to estimate normalised values of temporal expressions using different kinds of temporal units [Kolomiyets, 2012]. In TIDES, the <TIMEX2> tag is intended to support a variety of applications, and temporal expressions are considered stand-alone targets to be annotated and extracted [Verhagen, 2004].

Table 2.5 desbribes the attributes of <TIMEX2> tag:

Examples:

```
<TIMEX2 VAL="2001-12-31">Last day of 2001</TIMEX2>
<TIMEX2 VAL="2014-03-21">today</TIMEX2>
```

Table 2.5: <TIMEX2> tag attributes as specified in TIDES [Kolomiyets, 2012].

Attribute	Function	Example
VAL	Normalised value	VAL='2010-01-20'
MOD	Temporal modifier	MOD='APPROX'
ANCHOR_VAL	Normalised form of the anchoring time expression	ANCHOR_VAL='2010-01-21'
ANCHOR_DIR	Relative direction between VAL and ANCHOR_VAL	ANCHOR_DIR='BEFORE'
SET	Used for expressions denoting sets of times	SET='YES'
COMMENT	Annotator's comment	COMMENT='autogenerated'

TimeML

TimeML[2] [Pustejovsky et al., 2003a] is an expressive language for temporal information annotation, designed to connect the processes of temporal analysis of a text with a representation and formal meaning of time. It is a specification language for event and temporal expressions in natural language text able to capture distinct phenomena in temporal markup, to anchor events to temporally denoting expressions, and to order relative event expressions.

TimeMLis a metadata standard scheme for markup of events and their temporal anchoring, being able to link an event to a time, and recognizing some temporal adverbials, such as temporal prepositions (e.g. “for", “during", “on", “at") and connectives (e.g. “before", “after", “while") [Mani, 2003]. As a general annotation scheme, TimeML provides an XML-compliant markup language and annotation scheme for times and events, capable of capturing all salient temporal information in a text [Verhagen, 2004].

TimeML captures temporal semantics in text, focused on systematic anchoring events to the times, and their relative order to each other. TimeML adopted the core of the STAG and remained compliant to the TIDES time expression annotation, keeping the notions of temporal object and temporal relation as central points in TimeML. Temporal objects express: time expression and events, marked up with the <TIMEX3> and <EVENT> tags respectively [Kolomiyets, 2012].

TimeML is based on four major tags:

- <TIMEX3> was introduced for annotating temporal expressions in text, extending the TIDES <TIMEX2> attributes;

[2] http://timeml.org/

- <EVENT> is used for annotating events and states in text, comprising tensed and untensed verbs, nominalisations, adjectives, predicative clauses and prepositional phrases;
- <SIGNAL> annotates textual elements used to make relations holding two temporal elements, such as temporal prepositions and conjunctions, prepositions signalling modality (“to"), and special characters (“-" and “/") that can denote ranges.
- <LINK> enables encoding different types of relations between temporal elements to establishing temporal ordering: BEFORE, AFTER, INCLUDES, IS_INCLUDED, DURING, DURING_INV, SIMULTANEOUS, IAFTER, IBEFORE, IDENTITY, BEGINS, ENDS, BEGUN_BY, ENDED_BY

TimeML distinguishes three kinds of temporal links used to encode temporal relations between events and time expressions: a) <TLINK> encodes temporal relations proper, b) <ALINK> encodes aspectual relations, and c) <SLINK> encodes modality, negation and factuality. Instead of annotating such temporal relations on the event itself, they are annotated in a separate non-input-consuming tag that links events and time expressions to each other [Verhagen, 2004].

Example: *Paul taught on Friday*

```
Paul <EVENT eid=e1 class=OCCURRENCE>taught</EVENT>
on <TIMEX3 tid=t1>Friday</TIMEX3>
<TLINK eventInstanceID=ei1 relatedToTime=t1 relType=is included/>
```

In TimeML, TLINKs have a relation type attribute *relType* valued with one of fourteen different relation types described in Table 2.6. Such relations are intended to be mutually exclusive, but the guidelines do acknowledge that especially the simultaneous relation can be a bit fuzzy [Verhagen, 2004].

Table 2.6: TimeML <TLINK> relation types [Verhagen, 2004].

Relation Type	Description
simultaneous	Events that happen at the same time or so close that distinguishing their times makes no temporal interpretation difference
before,after	Used for temporal precedence of events and times
ibefore,iafter	One event is immediately before or after the other
includes,is includes	For relations between the temporal expression and the event
holds,held by	Like the simultaneous relation, differing by the fact that they are relations between an event and a particular time
begins,begun by	A relation between one event and the start time of a period
ends,ended by	A relation between one event and the end time of a period
identity	Annotated as a tlink even though it is not a temporal relation proper

The Unknown relation was added to be used when it is often not possible to specify a temporal relation between two random events in a text, and the user is forced to provide a temporal relation then, making a distinction between relations that have not yet been considered by the annotator and relations that were considered but have no value. Furthermore, TimeML has no *Overlap* relation, motivated by the observation that this relation does not naturally occur in real texts [Verhagen, 2004].

2.2.3 Examples of Temporal IE Systems and Application

Although most of the proposed solutions described below are able to extract (or at least identify) imprecise temporal expressions, they do not perform arithmetic or logic operations with them, which precludes, for example, listing imprecise-temporal-based events chronologically.

[Schilder and Habel, 2003] describes a semantic tagging system that deals with temporal and event extraction. However, when trying to represent temporal information in a time domain, the system anchors the temporal information obtained from natural language expressions in absolute time, i.e. in a linearly ordered set of abstract time-entities. However, this seems not to be very suitable to handle subjective expressions, such as “the beginning of next year".

TIE [Ling and Weld, 2010] is an IE system which distils facts from text, performing inference to bound the start and ending times for each event. TIE recognizes a wide range of temporal expressions and runs probabilistic inference to extract constraints on the endpoints of event-intervals.

LX-TimeAnalyzer [Costa and Branco, 2012] is a temporal analyser for Portuguese capable of fully annotating raw text with temporal information. LX-TimeAnalyzer uses a trained classifier to identify $type$ and $value$ of temporal expression, including: (a) word tokens composing the temporal expression, (b) temporal expression anchor often required for normalisation (e.g. “the following day" needs an anchor to be normalised), and (c) the broad tense ($present$, $past$, or $future$) used to decide whether an expression like “February" refers to the previous or the following month of February.

4D Fluents [Batsakis and Petrakis, 2011] is an approach for handling temporal knowledge in OWL ontologies, representing qualitative (temporal relation extensions, such as “before" and “after") and quantitative temporal information (where temporal information is defined precisely, e.g. using dates and times) through time instants and temporal intervals. In this approach, OWL concepts varying in time are represented as 4-D dimensional objects (4^{th} dimension being the time). Properties having a time dimension are called fluent properties, and domain (and range) of such type of properties is a class *TimeSlice*.

[Zhou et al., 2005] proposes an architecture able to process and discovering implicit temporal information in clinical narrative reports, which includes a post-processor that resolves temporal expressions and deals with issues such as granularity and vagueness. The proposed system encodes assertions using granularity stated in the text (e.g. a year is mapped to 365.2425 days). However, even a expression like “exactly one year ago" cannot be necessarily computed as something that happened 365 days, 5 hours, 49 minutes and 12 seconds ago.

[Goralwalla et al., 2001] provides an approach to the treatment of granularity in temporal data extraction, motivated by the known problem of diagnosing and following up unstable angina based on patient cardiologic problems. Angina is a transitory clinical syndrome usually associated with symptoms related to coronary artery disease. In this problem, it is important to consider both the time when the symptoms (like chest pain) began and the time duration of these symptoms. Figure 2.5 lists an example of information contained in the cardiologic medical record of a patient.

A system that analyses the content of medical records, for example, should be ready to extract and understand a temporal expression that represents a period of time, such as “*From December 1994 to April 1996* the patient took aspirin" (line 6 in Figure 2.5). Otherwise, this system will not be able to answer questions like “who used aspirin for more than 3 consecutive months?".

Table 2.7 illustrates how IE must be able to derive some extra information from the stored sentences about the patient to support query answering, and how IE systems must deal with temporal information and its granularity.

1:	The patient suffered from chest pain at rest for 2 hours and 55 minutes on 13 December 1995.
2:	The patient presented an episode of acute chest pain on 29 January 1996 from 13:20:15 to 13:56:23.
3:	The patient has been admitted to an Intensive Care Unit from 21:00 29 January 1996, and he has undergone intensive medical management for 36 hours.
4:	On 15 February 1996 the patient had myocardial infarction.
5:	At 3 pm 12 April 1996 the patient presented a new episode of chest pain of 7 minutes and 35 seconds during a soft exertion.
6:	From December 1994 to April 1996 the patient took aspirin.
7:	From 30 January 1996 the patient had to take a thrombolytic therapy for 38 months.

Figure 2.5: An example of a cardiologic medical record [Goralwalla et al., 2001].

KUL [Kolomiyets and Moens, 2013] is a system for temporal processing of text, which employs a number of machine learning classifiers to identify time expressions and events (“markables"), recognize their attributes, and estimate temporal links between recognized events and times. KUL's temporal parsing identifies temporal relation pair (event-event and event-timex) and the semantic label of the relation as a single decision, without specifying in advance the context in which these pairs have to occur. KUL recognizes temporal expressions by first pre-processing the input text (sentence detection, tokenization, part-of-speech tagging and parsing) using the OpenNLP package[3]. Second, temporal expressions candidates are selected according to the TIDES standard [Ferro et al., 2005], considering nouns (week, day), proper names (Tuesday, May), noun phrases (last Tuesday), adjectives (current), adjective phrases (then current), adverbs (currently), adverbial phrases (a year ago), and numbers (2000) [Kolomiyets and Moens, 2010]. Special labels for single tokens are used to detect parts of temporal expressions that cannot be found in the chunk-based fashion. Further, Timex classifiers are trained with a feature-vector extracted for phrase-candidate which includes: a) the head word of the phrase and its POS tag; b) all tokens and POS tags in the phrase as a bag of words; c) the word-shape representation of the head word and the entire phrase, e.g. “Xxxxx 99" for the expression “April 30"; d) the condensed word-shape representation for the head word and the entire phrase, e.g. “X(x) (9)" for the expression “April 30"; e) the concatenated string of the syntactic types of the children of the phrase in the parse tree; and f) the depth in the parse tree. Normalisation of temporal expressions estimates standardized temporal values and types (DATE, TIME, DURATION and SET), using a manually constructed categorized vocabulary. Each entry specifies a value of a temporal field, a final date/time value, or a method with parameters to apply. Vocabulary comprises the following categories: ordinal numbers, cardinal numbers, month names, week day names, season names, parts of day, temporal directions, quantifiers, modifiers, approximators, temporal co-references, fixed single token timexes, holidays, temporal units, and fine-grained categories introduced (day number, month number and year number).

NavyTime [Chambers, 2013] is a system used to identify event order from raw text. It first extracts events using contextual features and a rule-based extractor for time expressions. Events and times are linked by identifying ordered pairs, and labelling the ordering relations. Raw text presents a challenge of extracting the relevant pairs before labelling them. After a

[3] http://opennlp.apache.org/

Table 2.7: Dealing with temporal information [Goralwalla et al., 2001].

#	Question	Solution
1	What is the time span between the myocardial infarction and the last episode of chest pain?	To derive this, we need to compute the elapsed time (which is a time span) between the time instants 15 February 1996 and 3 pm 12 April 1996 (see sentences 4 and 5).
2	What is the global span of the symptoms of angina?	To answer this question, the elapsed time between the time instants 13:20:15 29 January 1996 and 13:56:23 29 January 1996 has to be added to the time spans 7 minutes and 35 seconds, and 2 hours and 55 minutes (see sentences 1, 2, and 5).
3	When did the patient finish the intensive medical management and what is the time span between the end of the intensive medical management and the onset of the new angina episode?	To answer the first part of the question, we need to add the time span 36 hours to the time instant 21:00 29 January 1996. The elapsed time between the resulting time instant and the time instant 3 pm 12 April 1996 gives the answer to the second part of the question (see sentences 3 and 5).
4	Was the patient taking aspirin when the past episode of chest pain happened?	The answer to this question depends on what interpretation we choose to give to the temporal labels December 1994 and April 1996. If we consider that the patient took aspirin from sometime in December 1994 to sometime in April 1996, then we cannot give a definite answer to the question. However, if we interpret December 1994 and April 1996 to mean the entire specified months, then December 1994 means the entire period between 00:00:00 1 December 1994 and 23:59:59 31 December 1994. Similarly, April 1996 means the entire period between 00:00:00 1 April 1996 and 23:59:59 30 April 1996. In this case we are able to give a definite answer that the patient was taking aspirin when the episode of chest pain happened (see sentences 5 and 6).
5	When did the thrombolytic therapy end?	In this case we have to add the time span 38 months to the time instant 30 January 1996 (see sentence 7).

relation is identified, it is labelled based on the result of 5 independent classifiers (AFTER, BEFORE, etc.), which yield better performance than the traditional 3 (or even 1). NavyTime uses SUTime [Chang and Manning, 2012], a rule-based system that extracts phrases and normalises them to a TimeML time. SUTime was improved with TimeBank specific rules, which makes NavyTime outperformed SUTime by over 3.5 points on time normalisation. NavyTime identifies relations using two different approaches (rule-based and data-driven) and then apply a traditional ordering task to label such relations (TempEval-3 uses the full set of 12 relations).

ClearTK-TimeML [Bethard, 2013] is a pipeline system which uses machine-learning models with a small set of simple features to predict temporal relations for a small set of syntactic, restricting temporal relation classification to a subset of constructions and relation types for which the models are most confident. Features used by the classifiers are derived from either tokens, part-of-speech tags or syntactic constituency parses. ClearTK was ranked 1^{st} in relation, time extent strict and event tense accuracy, during the TempEval 2013 event. To identify a time extent, each token is classified as being part of the B(eginning) of, I(nside) of, or O(utside) of a time expression, based on a set o features used to characterize tokens, which includes for each token: a) text; b) stem; c) part-of-speech; d) the unicode character categories for each character of the token, with repeats merged (e.g. Dec28 would be 'LuLlNd'); e) the temporal type of each alphanumeric sub-token, derived from a 58-word gazetteer of time words; and f) all of the above features for the preceding 3 and following 3 tokens. Time type identification is faced as a multiclass classification task. Each time is classified as DATE, TIME, DURATION or SET, using the following features: a) the text of all tokens in the time expression; b) the text of the last token in the time expression; c) the unicode character categories for each character of the token, with repeats merged; and d) the temporal type of each alphanumeric sub-token, derived from the same 58-word gazetteer of time words used to identify time extents.

HeidelTime [Strötgen et al., 2013] is a multilingual temporal tagger extraction and normalisation of temporal expressions. HeidelTime's existing English resources were tuned to develop new resources for the Spanish language, achieving the best results among all participants in task A in the TempEval-3 challenge for both languages. HeidelTime is a rule-based system which uses a strict separation between source code and language-dependent resources. While source code includes the strategies for processing different domains, resources consist of different files types that are read by the HeidelTime's well-defined rule syntax interpreter:

1. Pattern files: contain words and phrases typically used to express temporal expressions (e.g. names of months);

2. Normalisation files: include normalisation information about each pattern (e.g. the value of a specific month's name);

3. Rule files: include rules for date, time, duration, and set expressions.

[Fagerberg, 2014] presents an extraction process that uses a sorted hierarchy of regular expressions to recognize temporal expressions. Such hierarchy defines the precedence specific rules take over the more general ones, in order to avoid fault matches when evaluating similar expressions – e.g. "*two weeks*" (duration) and "*two weeks ago*" (a point in time). Extracted expressions are categorised according to a "type" (category) associated to each regular expression. Normalisation is provided by a function that transforms the matched expression into a normalised form using `<TIMEX3>` tags. To produce a valid result, normalisation functions can vary in complexity, from a simple rewrite to requiring additional calculations and context about the input source (e.g. publication date of the document). Such approach is implemented as a 5 words-based sliding window that is used to evaluate if temporal expressions exist within such part of the input text. As matching process fails, last word is removed from the current window until the window becomes empty. The window is moved one step if no matches are found. When a n-word valid match is found the following $(n-1)$ window movements are discarded. It aims to avoid the matching of smaller expressions inside an already matched larger expression.

[Bethard et al., 2007] presents an approach to produce knowledge representations by automatically extracting a timeline structure from text, aiming to identify temporal and causal relations which tie the identified events. Such automatically extracted timeline summarizes the information necessary to answer questions like "what happened first, A or B?". An extension of the TimeBank[4] and TempEval annotation guidelines was used to selected pairs of events from the the Wall Street Journal section of the TimeBank corpus. Such pairs of events were manually annotated them with the labels BEFORE, OVERLAP and AFTER, and the resulting corpus was used to train a support vector machine (SVM) model which could identify new temporal relations with 89.2% accuracy.

In [Llorens et al., 2012], authors use the argument that temporal expression normalisation can only be effectively performed with a large knowledge base and set of rules, to present a novel tool for temporal expression normalisation (TIMEN), in order to be a high-performance multi-lingual timex normalisation system, and a normalisation system that can be made permanent, reusable and extensible. TIMEN normalisation approach consists in converting the timex phrase together with some contextual into a symbolic representation using a knowledge base (KB). Then, such representation is matched against a set of rules to produce a normalised output in TIMEX3 format.

[Sun et al., 2013] presents an overview of the state of the art in clinical natural language processing (NLP) on text-based temporal reasoning in clinical informatics, focused on the

[4] `http://timeml.org/site/timebank/timebank.html`

temporal information represented in the unstructured narratives of clinical notes. As an essential dimension for the interpretation of clinical narratives, time provides a context that makes meaningful the order in which the symptoms develop, the timing of different treatments, and the duration and frequencies of medications.

Table 2.8 depicts the approaches and features found in the temporal information extraction systems previously mentioned.

Table 2.8: Common approaches and features used Temporal IE systems.

Reference	Approach	Feature Assignments	Temporal Relations	Temporal Reasoning	Vague References	Granularity Conversion	Event Ordering
[Schilder and Habel, 2003]	Semantic Tagging	X	X				
[Ling and Weld, 2010]	Probabilistic Inference	X	X				
[Costa and Branco, 2012]	Machine-learning Classifier	X					
[Batsakis and Petrakis, 2011]	Ontology Model		X	X			
[Zhou et al., 2005]	Framework	X		X	X		
[Goralwalla et al., 2001]	Reasoning Model			X		X	
[Kolomiyets and Moens, 2013]	Machine-learning Classifier	X	X		X		
[Chambers, 2013]	Rule-based Extractor	X	X	X	X		X
[Bethard, 2013]	Machine-learning Model	X	X		X		
[Strötgen et al., 2013]	Rule-based Extractor	X					
[Fagerberg, 2014]	Regular Expression Hierarchy	X					
[Bethard et al., 2007]	Knowledge Representation	X	X				
[Llorens et al., 2012]	Rule-based Extractor	X			X		

2.2.4 Temporal Fuzzy Logic

Time modelling is an important feature in many application domains, as historical information, temporal information is often uncertain, subjective and vague. Uncertainty relies on events which have contradictory facts stated from different source documents, e.g. disagreement over which time specification is the right one for someone's birthdate. Subjectivity can be observed on events related to named historical periods, e.g. "industrial revolution", which do not have a clear definition, so it is impossible to clearly state exactly when such events occurred. Vagueness states about events that are fuzzily defined (some, few, many) over different time granularity (days, weeks, months) [Nagypál and Motik, 2003].

Considerable effort has been carried out to extract temporal information from natural language texts, allowing question answering systems to deal with more complex temporal questions. However, temporal relationships expressed in natural language are often vague (which is inherently associated with real-world temporal information), and it is necessary to extend traditional temporal reasoning formalisms to cope with this kind of vagueness [Schockaert et al., 2008].

In temporal question answering systems, the boundaries of time periods can often be vague. In such cases, it is necessary to represent time periods with a formalism that is tolerant of imprecision and uncertainty. Answering a complex question may require decomposing the original question into partial questions, to answer such partial questions and combine the partial answers into the final answer. Temporal questions are an important class of complex questions, in which the accurate representation of the time span of events is essential to the treatment of such complex questions [Schockaert, 2005].

However, a lot of time information is ill-defined, subjective or uncertain. Thus, the time span representation should be tolerant of imprecision in temporal question answering systems. The fuzzy set theory is a representation formalism suitable for this purpose, allowing the definition of a gradual beginning and ending of events [Nagypál and Motik, 2003].

A fuzzy set is the basic concept that underlies the fuzzy systems theory [Pedrycz and Gomide, 1998], and involves capturing, representing, and working with linguistic

notions, being employed in those circumstances where impreciseness, unpredictability, and vagueness are in concern.

Definition 2.2 (Fuzzy Set) *A fuzzy set S is characterized by a membership function A mapping the elements of a (finite or not) domain, space or universe of discourse T into the unit interval* [0, 1]. *That is,* $A(t) : T \rightarrow [0, 1]$ *[Zadeh, 1994].*

A membership function A can be defined in different forms, such as triangular or trapezoidal functions, or continuously differentiable curves with smooth transitions, such as normalised Gaussian functions. The *height* of a fuzzy set S is the largest membership grade of any element in that set (Equation 2.1), whereas a fuzzy set S is called *normal* when $height(S) = 1$, and *subnormal* otherwise [Pedrycz and Gomide, 1998].

$$height(S) = max\{A(t), t \in T\} \tag{2.1}$$

The *support* of S, $supp(S)$, is the crisp set with all the elements of T satisfying $A(t) > 0$. Likewise, the *core* of S, $core(S)$, is the crisp set with all the elements of T satisfying $A(t) = 1$, whereas its *boundary*, $bound(S)$, encompasses all the elements of T with membership grades in the range]0, 1[, as shown in Figure 2.6 [Coelho and Raposo, 2005].

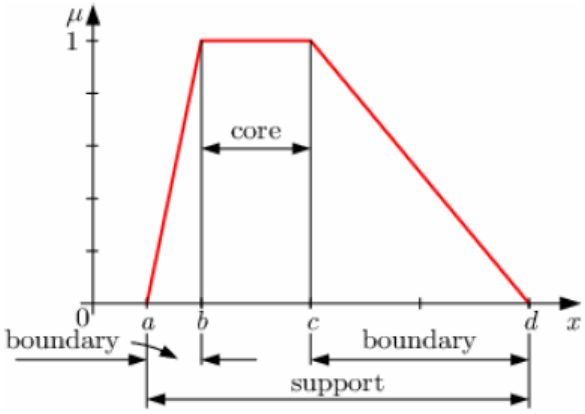


Figure 2.6: Concepts related to a fuzzy set [Coelho and Raposo, 2005].

Examples of Fuzzy Temporal Applications

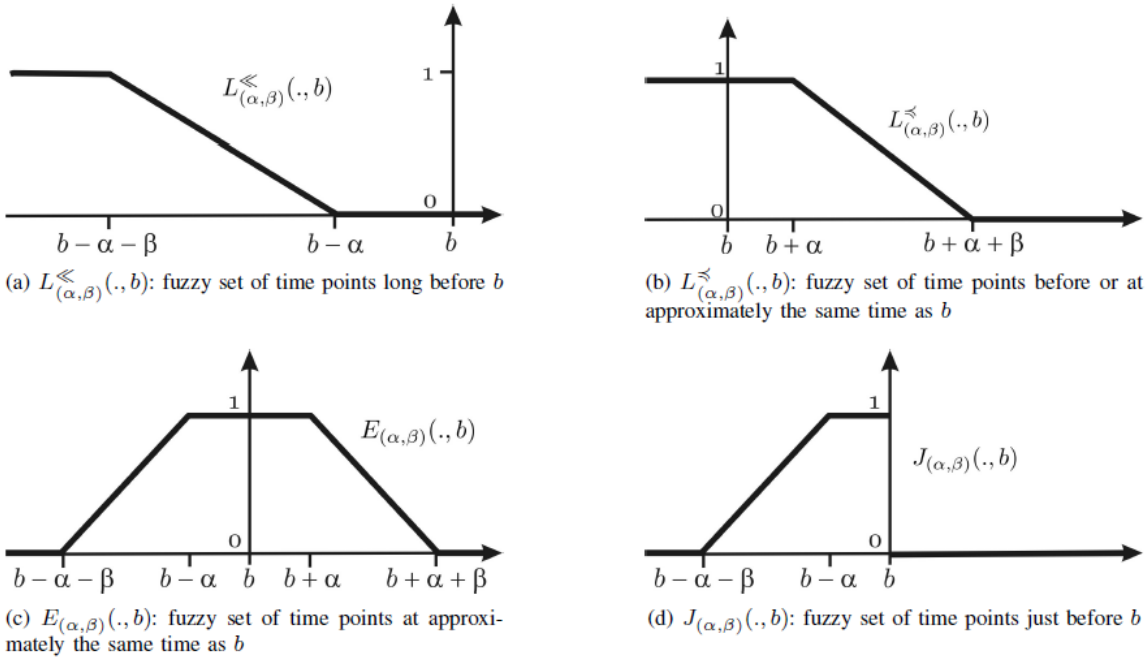
In [Nagypál and Motik, 2003], a fuzzy interval-based temporal model capable of representing imprecise temporal knowledge is described. It generalises Allen's temporal relations on intervals, by providing a definition of crisp interval relations based on set theory and then generalised them to the fuzzy case. The presented temporal model is intended for use in ontology modelling, following a modular semantics pattern which tries to keep the semantics of each model separate and to provide clean interfaces between them. Examining the different properties of the fuzzy temporal relations (like transitivity), one can observe basic inferences even in case of fuzzy intervals.

[Schockaert et al., 2008] presents a framework to represent, compute and reason about temporal relationships between events that have imprecise time spans, represented by fuzzy sets (*fuzzy time interval*). The proposed model preserves many of the Allen's relations properties, and it uses a transitivity table for efficient fuzzy temporal reasoning, with two main concerns:

1. The definitions of Allen's qualitative relations must be generalised to make them applicable not only to crisp intervals, but also to fuzzy intervals. Such fuzzy (time) intervals describe events by a gradual beginning and/or ending, and they can be defined either by an expert [Nagypál and Motik, 2003], or it can be constructed automatically [Schockaert, 2005].

2. To provide a means to model imprecise relations, being able to express different types of relations between, for instance, *event A took place just before event B*, and that *A occurred long after B*.

The qualitative relations between two fuzzy intervals are defined in terms of the ordering of the gradual beginning and endings of these intervals (ordering of the time points belonging to these intervals). Four basic fuzzy relations are defined to order 2 time points a and b. Parameters α and β are used to define such fuzzy set intervals, as shown if Figure 2.7.



(a) $L^{\ll}_{(\alpha,\beta)}(.,b)$: fuzzy set of time points long before b

(b) $L^{\preccurlyeq}_{(\alpha,\beta)}(.,b)$: fuzzy set of time points before or at approximately the same time as b

(c) $E_{(\alpha,\beta)}(.,b)$: fuzzy set of time points at approximately the same time as b

(d) $J_{(\alpha,\beta)}(.,b)$: fuzzy set of time points just before b

Figure 2.7: Fuzzy ordering of time points [Schockaert et al., 2008].

a) a occurs long before b;
b) a occurs before or at approximately the same time as b;
c) a occurs at approximately the same time as b;
d) a occurs just before b.

To order vague boundaries, fuzzy ordering relations are used to measure the extent to which the beginning of a fuzzy time interval A is long before the beginning of a fuzzy time interval B, taking into account the highest extent to which there exists a time point in A that occurs long before all the time points in B.

Examples of Fuzzing Temporal Boundaries

[Schockaert, 2005] suggests an approach based on fuzzy sets to define the beginning and ending of events, and provides a fully automatic procedure which uses statements on the web to construct the membership functions. To obtain useful statements from the web, authors used

the snippets returned by Google[5] for some automatically generated queries. In most applications, all membership functions are defined by an expert. However, this is considered the first attempt to construct membership functions for fuzzy time periods in an automatic way. The method is described in the following three parts.

Part 1: Creating fuzzy set boundaries

1. Let T be a linear ordered set of time points. For an event x, two fuzzy sets X_b and X_e (Equations 2.2 and 2.3) in T will be construct to represent the temporal beginning and end of event x.

2. For a point in time t in T, $X_b(t)$ expresses to what extent t is after or equal to the beginning of x, and $X_e(t)$ expresses to what extent t is before or equal to the end of x.

3. The time span for x is $X = X_b \cap X_e$.

4. Let $b_1, b_2, ..., b_n$ be the possible beginnings for x that were extracted from the web ($b_i \in T$) and let $f(b_i)$ be the number of times b_i was found as a possible beginning ($1 \leq i \leq n$). Analogously, let $e_1, e_2, ..., e_m$ be the possible endings for x ($e_i \in T$) and let $g(e_i)$ be the number of occurrences of e_i as a possible ending ($1 \leq i \leq m$).

5. Each occurrence of a possible beginning b_i is a test subject that would answer affirmative to the question “Is t after or equal to the beginning of x?" for all $t \geq b_i$, and negative for all $t < b_i$. Each occurrence of a possible ending is treated in a similar way.

6. X_b and X_e are defined for t in T as:

$$X_b(t) = \frac{\sum_{b_i \leq t} f(b_i)}{\sum_{i=1}^{n} f(b_i)} \tag{2.2}$$

$$X_e(t) = \frac{\sum_{e_i \geq t} g(e_i)}{\sum_{i=1}^{m} g(e_i)} \tag{2.3}$$

Two problems arises in such approach when using the web to obtain the data. First, the presence of inconsistent information on the web makes it not possible to assume that $b_i \leq e_j$ for all i in $\{1, 2, ..., n\}$ and j in $\{1, 2, ..., m\}$, and the resulting time span X would not be a normalised fuzzy set. Second, the beginnings and endings of an event can be defined by means of an interval or even by a vague description (e.g. “the late 1930s").

Part 2: Dealing with inconsistency

The problem of handling inconsistency is solved by discarding possible beginnings that come after all possible endings and possible endings that come before all possible beginnings, assuming without loss of generality that:

$$\max_{i=1}^{n}(b_i) \leq \max_{j=1}^{m}(e_j) \qquad \textit{and} \qquad \min_{i=1}^{n}(b_i) \leq \min_{j=1}^{m}(e_j)$$

In a second approach, possible beginnings (or endings) that cannot be separated by a possible ending (or beginning) are grouped, making that ambiguous events which took place at approximately the same time, may be clustered together.

[5] http://www.google.com

Let $B_1, B_2, ..., B_k$ be a partitioning of $\{b_1, ..., b_n\}$ and $E_1, E_2, ..., E_k$ be a partitioning of $\{e_1, ..., e_m\}$ such that $\max(B_i) < \min(B_i + 1)$ and $\max(E_i) < \min(E_i + 1)$ for all i in $\{1, ..., k-1\}$ and $\max(B_j) < \min(E_j)$ for all j in $\{1, ..., k\}$.

Some of the B_i's or E_j's may contain only noisy (incorrect) dates and a possible solution is to consider only groups of dates that correspond to a significant number of occurrences:

Let α be a small constant in $]0, 1[$. For all i in $\{1, 2, ..., k\}$, all dates in B_i and all dates in E_i are discarded when:

$$\sum_{b \in B_i} f(b) < \alpha \max_{1 \leq j \leq k} \Big(\sum_{b \in B_j} f(b) \Big) \qquad or \qquad \sum_{e \in E_i} g(e) < \alpha \max_{1 \leq j \leq k} \Big(\sum_{e \in E_j} g(e) \Big)$$

That results new groups $B_1^{'}, ..., B_{k'}^{'}$ and $E_1^{'}, ..., E_{k'}^{'}$ where some of the B_i's (and E_i's) are taken together – when B_i is discarded, then E_{i-1} and E_i are taken together. Finally, for each every pair $(B_i^{'}, E_j^{'})$ with $i < j$, the membership function X (as *before*) results the number of possible time spans for the event under consideration, and for each time span corresponding to a pair $(B_i^{'}, E_j^{'})$ can be assigned a score s_{ij} in $[0, 1]$ (Equation 2.4):

$$s_{ij} = \frac{\sum_{b \in B_i^{'}} f(b)}{\max_{1 \leq l \leq k'} \Big(\sum_{b \in B_i^{'}} f(b) \Big)} \times \frac{\sum_{e \in E_j^{'}} g(e)}{\max_{1 \leq l \leq k'} \Big(\sum_{e \in E_j^{'}} g(e) \Big)} \tag{2.4}$$

Part 3: Dealing with imprecision

An underspecified or vague date can be interpreted as: a) an event started on a particular date which there exists uncertainty about the exact date beginning (e.g. “December 2013"); or b) an event that began gradually during an underspecified or vague period (e.g. “in the end of last year").

In [Schockaert, 2005], both underspecified and vague dates are represent as fuzzy sets:

1. Let $\beta_1, \beta_2, ..., \beta_s$ be fuzzy sets in T representing the possible underspecified or vague beginnings of an event x, and $\epsilon_1, \epsilon_2, ..., \epsilon_r$ be fuzzy sets in T representing the underspecified or vague possible endings.

2. Let $f(\beta_i)$ be the number of occurrences of β_i as a possible beginning and let $g(\epsilon_j)$ be the number of occurrences of ϵ_j as a possible ending.

3. As before, let $b_1, b_2, ..., b_n$ and $e_1, e_2, ..., e_m$ be the possible fully specified beginnings and endings.

4. If exists some b_j, where $\beta_i(b_j) > 0$ for $(1 \leq i \leq s, 1 \leq j \leq n)$, the first interpretation (the event started on a particular date which is unknown) holds for β_i; otherwise, the event began gradually – the same rule is considered for possible endings.

5. Let the sets I_1, F_1 (Equations 2.5 and 2.6) be the set of beginnings and endings for which the first interpretation is assumed (dates which are unknown) and I_2, F_2 (Equations 2.7 and 2.8) be the set of beginnings and endings for which the second interpretation is assumed (dates that begin and end gradually), defined as:

$$I_1 = \{\beta | \beta \in \{\beta_1, ..., \beta_s\} \ and \ (\exists i \in \{1, ..., n\})(\beta(b_i) > 0)\} \tag{2.5}$$

$$F_1 = \{\epsilon | \epsilon \in \{\epsilon_1, ..., \epsilon_r\} \ and \ (\exists i \in \{1, ..., m\})(\epsilon(e_i) > 0)\} \tag{2.6}$$

$$I_2 = \{\beta_1, ..., \beta_s\} \setminus I_1 \tag{2.7}$$

$$F_2 = \{\epsilon_1, ..., \epsilon_r\} \setminus F_1 \tag{2.8}$$

6. If A represents an underspecified or vague date (A is a fuzzy set in T), the fuzzy sets $A^-(t)$ and $A^+(t)$ (Equations 2.9 and 2.10) in T for t in T express he extent to which t is after or this date, and are defined as:

$$A^-(t) = \frac{\int_{-\infty}^{t} A(x)dx)}{\int_{-\infty}^{+\infty} A(x)dx} \tag{2.9}$$

$$A^+(t) = \frac{\int_{t}^{+\infty} A(x)dx)}{\int_{-\infty}^{+\infty} A(x)dx} \tag{2.10}$$

7. The fuzzy sets X_b and X_e (Equations 2.11 and 2.12) are then defined for t in T as:

$$X_b(t) = \frac{\sum_{\beta \in I_2} f(\beta)\beta^-(t) + \sum_{b_i \leq t} f(b_i) + \sum_{\beta \in I_1} f(\beta)\frac{\sum_{b_i \leq t} \beta(b_i)f(b_i)}{\sum_{j=1}^{n} \beta(b_j)f(b_j)}}{\sum_{i=1}^{n} f(b_i) + \sum_{i=1}^{s} f(\beta_i)} \tag{2.11}$$

$$X_e(t) = \frac{\sum_{\epsilon \in F_2} g(\epsilon)\epsilon^+(t) + \sum_{e_i \geq t} g(e_i) + \sum_{\epsilon \in F_1} g(\epsilon)\frac{\sum_{e_i \leq t} \epsilon(e_i)g(e_i)}{\sum_{j=1}^{m} \epsilon(e_j)g(e_j)}}{\sum_{i=1}^{m} g(e_i) + \sum_{i=1}^{r} g(\epsilon_i)} \tag{2.12}$$

where in Equations 2.11 and 2.12: a) the first term in the numerator represents the influence of underspecified and vague dates for which the second interpretation is assumed (dates that begin and end gradually); b) the second term represents the influence of exact dates; and c) the third term represents the influence of vague and underspecified dates for which the first interpretation is assumed (particular dates which are unknown).

Figure 2.8 shows an example that considers the time span of the World War 2. There does not exist a unique point in time that corresponds to the beginning or ending of this war.

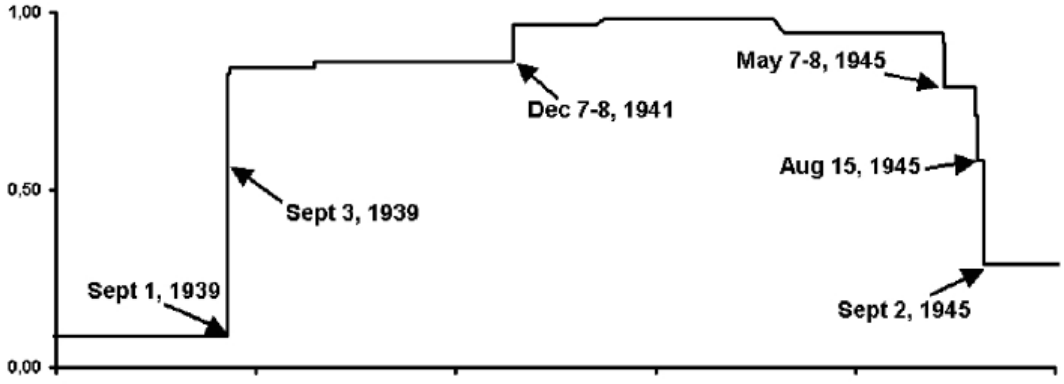


Figure 2.8: Fuzzy set representing the time span of World War 2 [Schockaert, 2005].

This proposal is based on the fact that the Internet provides a set of possible definitions for the temporal beginning and end of the analysed events. It uses a statistical approach to generate a membership function to define the fuzzy sets that will represent the beginning and end of each event. However, it does not provide a description of how each element of fuzzy sets $\beta_1, \beta_2, ..., \beta_s$ and $\epsilon_1, \epsilon_2, ..., \epsilon_r$ in T that represent possible underspecified or vague beginnings endings of an event x are defined, i.e. how the membership functions are defined for each β_i and ϵ_j.

A similar approach was used in [Blamey et al., 2013] to represent a temporal expression S by a function $f(t)$, which is a probability density function for the continuous random variable T_s, using photographs uploaded to the photo-sharing site Flickr[6]. After collecting a list of timestamps for an specific temporal term, the target is to find a probability density function to provide a convenient representation, and smooth the data appropriately. Author argues that temporal expressions can communicate more than points and intervals, and their cultura meaning is much more complex – often difficult to be precisely defined. Thus, a distributed definition can capture such cultural meaning in a more detailed way, as shown in Figure 2.9 for the expression “Christmas".

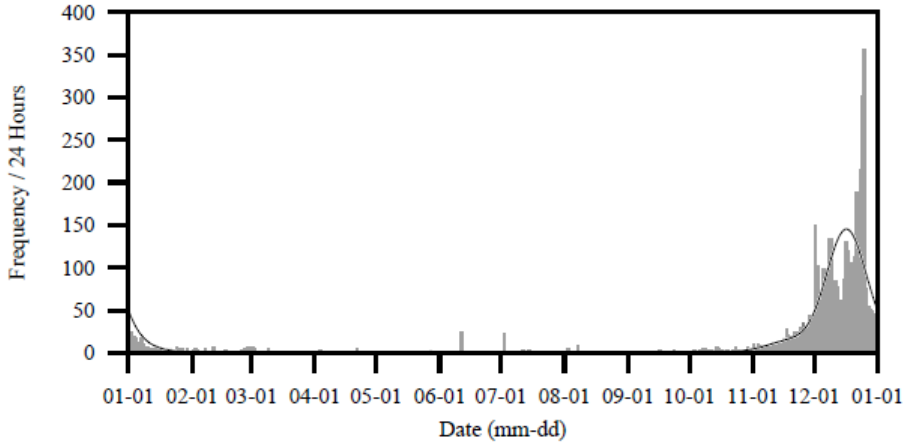


Figure 2.9: Distribution of “Christmas" images on Flicr [Blamey et al., 2013].

2.3 Dealing with Spelling Errors in IE

In IE, existing approaches use methods to search for valid words within a text, having a dictionary as support. However, they have two main drawbacks. First, the known solutions may be inefficient in the presence of spelling errors. Second, the existing dictionaries are not rich enough to encode phonetic information to assist the search.

Given a string set and a query string, the string similarity search problem is to efficiently find all strings in the string set that are similar to the query string. In a full search, a given word is compared to all words in the repository to result in a set of similar words. Regardless of the greater search time, it is guaranteed that all the similar words that satisfy the minimal similarity criteria will be returned. In a fast search, we want to decrease the search time by reducing the number of words to be compared. However, not all words that satisfy the minimum similarity criteria may be returned [Fenz et al., 2012b].

[6] http://www.flickr.com

The extraction of unstructured data coupled with a supporting dictionary can be very inefficient, in particular when the analysed text has spelling errors [Stvilia, 2007]. String similarity methods typically do not necessarily cover the specific application aspects related to spelling errors. In these cases, it is necessary to use phonetic similarity metrics. Phonetics are language-dependent [Mann, 1986] and solutions for this sort of problems must be specially designed for each specific language. In addition, similarity algorithms are often slow when executed over large databases. Although some fast search algorithms have been implemented [Bocek et al., 2007], their results are based on string distance metrics and do not consider phonetic similarity.

2.3.1 Similarity Metrics and Search

As the presence of errors must be considered when analysing text in a IE process, string similarity is an important feature to be addressed.

Definition 2.3 (String Similarity) *String Similarity or String Distance is a metric that measures similarity or dissimilarity (distance) between two text strings for approximate string matching or comparison [Gomaa and Fahmy, 2013].*

Several proposed functions can be found in the literature to measure string similarity. One of the well known functions is Edit Distance (ED) (or Levenshtein Distance) [Levenshtein, 1966]. $ED(w_1, w_2)$ – calculates the minimum number of operations (single-character edits) required to transform string w_1 into w_2. ED can also be normalised to calculate a percentage similarity instead of the number of operations needed to transform one string to another. As examples of its application, [Álvarez et al., 2007] proposes a successfully technique for web data extraction using ED to calculate the similarity between two sequences of consecutive sibling subtrees in the tree of an html page, and [Heeringa, 2004] used the Levenshtein Distance to measure differences on dialect pronunciations over 27 Dutch dialects in a database. Another well-known string similarity metric is the Jaro-Winkler distance [Winkler, 1990], generally used to compare prefix of strings [Cohen et al., 2003]. [Gomaa and Fahmy, 2013] presents a survey with the existing works on text similarity. In addition, other examples of string similarity applications can be found in the literature:

- Hamming Distance [Hamming, 1950] calculates the number of bits (or characters) that are different between two vectors (or strings).
- Longest Common Subsequence [Paterson and Dancik, 1994, Allison and Dix, 1986] (LCS) finds the longest subsequence of two strings that is as long as any other common subsequence. ROUGE-L is an automatic method for machine translation evaluation based on LCS. Empirical results showed that the method is correlated with human judgments [Lin and Och, 2004]. AckSeer is “a search engine and a repository for automatically extracted acknowledgments in the CiteSeerX digital library", using LCS to evaluate disambiguation of abbreviations in the proposed dataset [Khabsa et al., 2012].
- Smith-Waterman [Smith and Waterman, 1981] distance was originally designed to identify similarities between linked DNA and protein sequences. The Smith-Waterman algorithm compares segments of all possible lengths and optimises the similarity measure. In [Su et al., 2008] we can find an application that uses the Smith-Waterman algorithm and Levenshtein Distance to detect plagiarism in academic papers. Monge-Elkan distance [Monge and Elkan, 1996] is a recursive variant of the Smith-Waterman distance function

which assigns a relatively lower cost to a sequence of insertions and deletions to identify equivalent data in multiple sources (“field matching problem").

- [Frozza and dos Santos Mello, ez06] adopted the Jaro-Winkler distance to compare the similarities of Geography Markup Language (GML) nodes and ontology tree node.
- [Cohen et al., 2003] compares different string distance metrics for name-matching tasks, including edit-distance like functions, token-based distance functions and hybrid methods, concluding Monge-Elkan distance [Monge and Elkan, 1996] performed best among several metrics.

In addition to the string distance metrics, the phonetic representation of words can be used to measure the phonetic similarity between them. String distance measures tend to ignore the relative likelihood errors. However, phonetic distances are able to assign a high score even though comparing dissimilar pairs of strings that produce similar sounds [Droppo and Acero, 2010].

Soundex [Hall and Dowling, 1980] is a phonetic matching scheme initially designed for English that uses codes[7] based on the sound of each letter to translate a string into a canonical form of at most four characters, preserving the first letter [Zobel and Dart, 1996]. For example, “reynold" and “renauld" are both reduced to "r543“. As the result, phonetically similar entries will have the same keys and they can be indexed for efficient search using some hashing method. However, Soundex fails to consider only the initial portion of a string to generate the phonetic representation, which impairs the phonetic comparison when words have more than 4-5 consonant phonemes. More commonly, Soundex also makes the error of transforming dissimilar-sounding strings such as “catherine" and “cotroneo" to the same code, and of transforming similar-sounding strings to different codes. The Soundex algorithm is given in Figure 2.10.

```
1: Replace all but the first letter of the string by its
   phonetic code;
2: Eliminate any adjacent repetitions of codes;
3: Eliminate all occurrences of code 0 (that is, eliminate
   vowels);
4: Return the first four characters of the resulting string.
```

Figure 2.10: Soundex algorithm [Hall and Dowling, 1980].

When dealing with a large repository in terms of volume of data, it also requires a structure to support a fast similarity search. It means that, for a given possible not well written word, we have to find similar words (based on certain similarity criteria), but not performing a full search in the repository.

State Set Index (SSI) [Fenz et al., 2012a] is an efficient solution for finding strings in a string set that are similar to the query string. SSI is based on a TRIE (prefix index) that is interpreted as a nondeterministic finite automaton and it implements a novel state labelling strategy making the index highly space-efficient.

Fast Similarity Search (FastSS) [Bocek et al., 2007] is an algorithm designed to find strings similarities in a large database. This algorithm is based on ED metric. According to the authors, in a dictionary that contains n words, and given a maximum number e of spelling errors, the FastSS algorithm creates an index of all n words containing up to e deletions. Each

[7] Soundex phonetic codes comprise: 0 for {a, e, i, o, u, y, h, w}; 1 for {b, p, f, v}; 2 for {c, g, j, k, q, s, x, z}; 3 for {d, t}; 4 for {l}; 5 for {m, n}; and 6 for {r}.

query is mutated, at search time, to generate a deletion neighbourhood, which is compared to the indexed deletion dictionary. The algorithm was tested and compared with NR-grep, a keyword tree, dynamic programming, n-grams, and neighbourhood generation using entries of the English Dictionary, English Wikipedia and a chapter from the book Moby Dick. The results show that FastSS performs faster than these algorithms. In [Bocek et al., 2008], the authors created a tailored FastSS to Peer-to-Peer domain entitled P2PFastSS, enabling a peer to search for a similar keyword in any text-based content, returning documents that contain similar keywords, ranking the result based on the Edit Distance. In [Bocek et al., 2009], P2PFastSS is used in mobile phones and laptop contexts, allowing users to publish and search for textual content containing misspellings. As disadvantage, however, FastSS has to manage a large database index set, and it does not consider phonetic similarity when trying to identify spelling errors.

2.3.2 Lexical Databases

A lexical database is a lexical resource that stores lexical category and synonyms of words, as well as semantic relations between different words or sets of words. A lexical database is associated to a software environment database which permits access to its contents, and such a database may be custom-designed for the lexical information or a general-purpose database into which lexical information has been entered. DANTE, EsPal and WordNet are some examples of lexical databases.

Dante is a lexical database for English which provides a fine-grained and comprehensive record of the behaviour of over 42,000 headwords and 23,000 multiword expressions, and a systematic description of the meanings, grammatical and collocational behaviour, and text-type characteristics of English words [Kilgarriff, 2010].

EsPal[8] is designed to be a source of information containing all the possible properties of Spanish words, created from a large collection of written data from the Web, government sources, newspapers, and literature.

Princeton WordNet (PWN) is the most commonly used computational lexicon of English for word sense disambiguation, a task aimed to assigning the most appropriate senses (i.e. synsets) to words in context [Navigli, 2009]. PWN is the English lexical database developed by the Cognitive Science Laboratory[9] at Princeton University and its database is divided by part of speech (noun, verb, adjective and adverb), and organized in sets of synonyms, called synsets, each of which representing the "meaning" of the word entry.

As a lexical database, PWN “is organized around the structure of synsets, sets of synonyms and pointers describing relations to other synsets" [Abderrahim et al., 2013]. As a semantic dictionary, PWN was designed to represent words and concepts as an interrelated system, consistent with evidence for the way speakers organize their own mental lexicons. PWN differs from other dictionaries, being neither a traditional dictionary nor a thesaurus, but it combines features of both types of lexical resources [Dantchev, 2013].

Figure 2.11 shows the main PWN repository entities and their relations. PWN contains synsets consisting of all the words that express a concept, and expressions that are related to this concept can be used to look up others. Synsets are linked with each other by numerous semantic relations, like hyponymys and meronymys. In addition, PWN gives definitions and sample sentences for most of its synsets. A definition is valid for all of the synonyms in the synset, since it expresses the sense of the combining concept. On the other hand, sample sentences are not the same for all synonyms [Dantchev, 2013].

[8] http://www.bcbl.eu/databases/espal/

[9] http://www.cogsci.princeton.edu

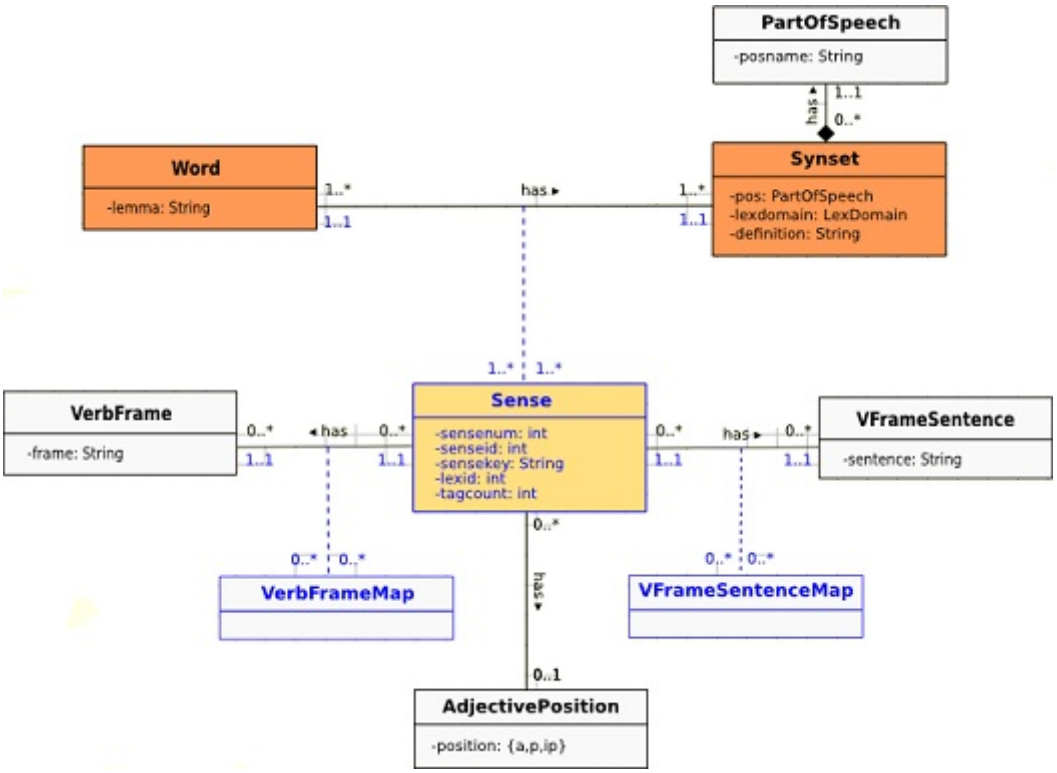


Figure 2.11: Main schema entities in the WordNet repository [Miller, 1995].

In spite of being designed for English, PWN is an inspiration in the development of this kind of lexical database, supporting several efforts to produce WordNet versions in other languages. It is an important resource in Natural Language Processing (NLP) applications, and also inter-linking WordNet of different languages to develop multilingual applications [Leenoi et al., 2009]. As an example, the Multilingual Central Repository (MCR 3.0) is a project that incorporates 5 different languages, such as English, Spanish, Catalan, Basque and Galician [Gonzalez et al., 2012].

PWN has a great acceptance in the academic body and it is established in IE researches. Although its widely used, PWN have some lacks and shortcomings that need to be improved to support IE applications, further in other language: (a) PWN contains a wide range of common words and it was designed to be an underlying database for different applications, not to cover special domain vocabulary, as need in IE systems, (b) unlike other dictionaries, PWN does not include information about derivative words and the forms of irregular verbs – this problem is even greater when considering the amount of verb conjugation variation in different tenses, e.g. 67 variations for each verb in Brazilian Portuguese [Ferreira, 2004] –, and (c) PWN should be extended to carry phonetic information in order to used it on searching for phonetically similar words.

2.4 Summary

In this chapter we presented the state of the art related to information extraction and temporal information extraction from text. First, we presented a brief description of IE systems and the most common features in a general IE architecture. We showed a simple pipeline architecture for an generic IE system, and we also compared features and components of some IE

systems to show those components can be combined in different ways to produce different IE solutions. We also summarized features and components found IE systems, showing that such systems do not usually deal with spelling errors. Secondly, we described the temporal concepts and the tasks that comprise the temporal information extraction process, highlighting some of the issues involved in such process. We showed how fuzzy logic can be used to describe some imprecise temporal concepts, and we pointed the challenges on extracting inaccurate temporal expressions from text. Finally, we explored the problem of dealing with spelling errors in the IE process, and how lexical databases can be integrated in the IE process to support similarity search.

Chapter 3

Imprecise Temporal Information Extraction and Normalisation

The extraction of temporal information from text is fundamental for language understanding [UzZaman and Allen, 2010]. Understanding temporal information is an important sub-task for several language processing applications, such as question answering, text summarisation, information retrieval [Derczynski et al., 2015], and knowledge base population [Burman et al., 2011]. Processing a temporal expression (timex) from text, i.e. extracting and modelling the expression, includes tasks such as recognition and representation of the temporal information [Kolomiyets, 2012].

In many situations, however, temporal expressions are not accurately described in the text. An imprecise timex denotes an imprecise amount or point in time, as in “less than a year", “a few days", and “recently". TimeML [Pustejovsky et al., 2003a] is the major initiative for temporal information annotation. TimeML provides a model and annotation scheme for temporal information in text, including the TIMEX3 scheme for representing temporal expressions. Although TimeML is capable of describing imprecise timexes in terms of language structure, the normalisation of imprecise temporal information in terms of values can be ambiguous or incomplete. For example, the TIMEX3 `mod` attribute allows for modification of expressions, but only in a very constrained way (twelve preset non-disjoint modifiers). Expressions like “some days" and “several days" are both normalised with the feature `VALUE="PXD"`, where “P" represents a period of time, “X" indicates an undetermined about of time, and “D" sets the temporal granularity as days.

Temporal information in some text types, e.g. clinical notes, can be imprecise, affecting for example the results of searches for events related to such temporal data. In addition, an inaccurate interpretation may yield different values for the same expression. For this reason, for a given application, it is important to estimate standardized values for the existing imprecise timexes, i.e., normalising them, giving a coherent view. Existing approaches use fuzzy sets to represent individual timexes and relations[Nagypál and Motik, 2003, Schockaert, 2005, Schockaert et al., 2008, Filannino and Nenadic, 2014]. However, they are focused on specific expressions or periods of time, and they do not provide a generic methodology for the normalisation of imprecise time expressions.

In this chapter, we discus different aspects of identifying and normalising temporal expressions extracted from text. In Section 3.1, we describe our participation in the recent SemEval-2015 Task 6 – Clinical TempEval[1] [Tissot et al., 2015a]. We developed two approaches

[1]This work has been published at SemEval-2015: Semantic Evaluation Exercises - International Workshop on Semantic Evaluation, Denver, Colorado, June 2015, co-located with NAACL-2015.

for timex identification. We describe a rule-based approach to time expression identification that we used in Clinical TempEval. A SVM-based approach is described in Appendix B. We discuss how they performed relative to each other, and how characteristics of the corpus affected outcomes and the suitability of the two approaches. In Section 3.2, we analyse the differences between our systems and the manually-annotated Clinical TempEval corpus, in order to discover the reason for the low precision reached by our rule-based approach[2] [Tissot et al., 2015b]. Our analysis demonstrated how difficult it is to create a manually annotated Gold Standard for time expressions and why this problem is still open in computational linguistics. The analysis is based on a methodology composed of six steps, from manual annotation of the input data, to finding and classifying the time expressions, and finally to a classification of the discrepancies found. Section 3.3 provides evidence in order to demonstrate the importance of dealing with imprecise temporal data within the IE process. In Section 3.4, we present a complete methodology for normalising imprecise time expressions, for a timex in situ and its coarse-grained TimeML class. This is the main contribution in this thesis. We perform a sequence of steps, from the pre-processing of the input data, to the application of statistical regression and machine learning techniques to produce trapezoidal and hexagonal membership functions that represent imprecise timexes. We compared the area of two membership functions to calculate a F1-score that guides the choice of the most suitable resulting membership function model. The result is a grounded probability density function for the period over which the timex was attained. In order to realise whether the differences are more concentrated at the top or at the bottom, we propose to use a complementary weighted variation of F1-score ($F1_{3D}$) in which each membership function is considered as a tridimensional object, varying the membership function depth from 0 at the bottom to 1 at the top. Section 3.5 provides practical experiments for imprecise timexes normalisation in both English and Portuguese languages.

3.1 Time Expression Identification in Clinical TempEval

SemEval (Semantic Evaluation) is a series of evaluations that aims to verify the effectivenesses of existing approaches to semantic analysis. Within the SemEval-2013 (Semantic Evaluation) workshop [UzZaman et al., 2013a], TempEval-3 was a competition for temporal expression, event, and temporal relation extraction, with the purpose to advance research on temporal information processing. Previous TempEval [Verhagen et al., 2009] and TempEval-2 [Verhagen et al., 2010] also included temporal annotation tasks, of which both were followed by informative analyses of the corpora and participant results, which led to a better understanding of the task as framed in these exercises [Lee and Katz, 2009, Derczynski, 2013]. The TempEval-3 Task “A" examined temporal information extraction and normalisation using the complete set of TimeML temporal relations.

IE systems are evaluated through precision, recall, and F1-score relevance measures. Precision is equivalent to the amount of retrieved instances that are relevant, while recall is equivalent to the amount of relevant instances that are retrieved. For classification and search tasks, the terms *true positives* (TP) and *true negatives* (TN) represent the correct result and the correct absence of results respectively, while the terms *false positives* (FP) and *false negatives*

[2]This work has been published at ISA-11: Eleventh Joint ACL - ISO Workshop on Interoperable Semantic Annotation, London, UK, April 2015, in conjunction with IWCS 2015.

(FN) correspond to the unexpected result and the missing result respectively. These terms are used to define Precision and Recall according to Equations 3.1 and 3.2:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

In addition, results are also presented in terms of F-measure (or F1-score), which is a measure of accuracy that considers both the precision and the recall to compute the score (Equation 3.3). F1-score result can be interpreted as the weighted average (or harmonic mean) between precision and recall, where F1-score reaches its best value at 1 and worst score at 0 [Davis and Goadrich, 2006]. The strict F1-score considers all partially correct responses as incorrect (spurious).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.3}$$

The three main tasks proposed for TempEval-3 focused on TimeML entities and relations were: a) Task A: Timex extraction and normalization, by determining the extent of the timexes in a text as defined by the TimeML, and also determining the value of the features TYPE and VALUE; b) Task B: Event extraction and classification, by detemining the extent of the events in a text as defined by the TimeML EVENT tag and the appropriate CLASS; and c) Task ABC: Temporal relation annotation, which entails performing tasks A and B, comprising: extract the temporal entities (events and timexes), identify the pairs of temporal entities that have a temporal link (TLINK), and classify the temporal relation between them, according to TimeML relation definition. Table 3.1 depicts the results obtained by the participant systems in SemEval-2013 TempEval-3 Task A.

In the recent SemEval-2015 Task 6, Clinical TempEval[3] [Bethard et al., 2015] was a temporal information extraction task over the clinical domain, using clinical notes and pathology reports for cancer patients provided by Mayo Clinic.[4] Clinical TempEval focused on identification of: spans and features for timexes, event expressions, and narrative container relations. The combined University of Sheffield/Federal University of Parana (UFPRSheffield) team focused on identification of spans and features for time expressions (TIMEX3) based on specific annotation guidelines (TS and TA subtasks). For time expressions, participants identified expression spans within the text and their corresponding classes: DATE, TIME, DURATION, QUANTIFIER, PREPOSTEXP or SET.[5] Participating systems had to annotate timexes according to the guidelines for the annotation of times, events and temporal relations in clinical notes – THYME Annotation Guidelines [Styler et al., 2014] – which is an extension of ISO TimeML [Pustejovsky et al., 2010] developed by the THYME project.[6] Further, ISO TimeML extends two other guidelines: a) TimeML Annotation Guidelines [Sauri et al., 2006], and b) TIDES 2005 Standard for the Annotation of Temporal Expressions [Ferro et al., 2005]. Clinical TempEval temporal expression results[7] were given in terms of Precision, Recall and F1-score for identifying spans and classes of temporal expressions. For Clinical TempEval two datasets were provided. The first was a training dataset comprising 293 documents with a total

[3] `http://alt.qcri.org/semeval2015/task6/`

[4] `http://www.mayoclinic.org`

[5] There was no `value` normalisation task in Clinical TempEval

[6] `http://thyme.healthnlp.org/`

[7] `http://alt.qcri.org/semeval2015/task6/index.php?id=results`

Table 3.1: TempEval-3 - Task “A" - Temporal Expression Performance [UzZaman et al., 2013b].

System	F1	P	R	Strict F1	Value F1
HeidelTime-t	90.30	93.08	87.68	81.34	77.61
HeidelTime-bf	87.31	90.00	84.78	78.36	72.39
HeidelTime-1.2	86.99	89.31	84.78	78.07	72.12
NavyTime-1,2	90.32	89.36	91.30	79.57	70.97
ManTIME-4	89.66	95.12	84.78	74.33	68.97
ManTIME-6	87.55	98.20	78.99	73.09	68.27
ManTIME-3	87.06	94.87	80.43	69.80	67.45
SUTime	90.32	89.36	91.30	79.57	67.38
ManTIME-1	87.20	97.32	78.99	70.40	67.20
ManTIME-5	87.20	97.32	78.99	69.60	67.20
ManTIME-2	88.10	97.37	80.43	72.22	66.67
ATT-2	85.25	98.11	75.36	78.69	65.57
ATT-1	85.60	99.05	75.36	79.01	65.02
ClearTK-1,2	90.23	93.75	86.96	82.71	64.66
JU-CSE	86.38	93.28	80.43	75.49	63.81
KUL	83.67	92.92	76.09	69.32	62.95
KUL-TE3RunABC	82.87	92.04	75.36	73.31	62.15
ClearTK-3,4	87.94	94.96	81.88	77.04	61.48
ATT-3	80.85	97.94	68.84	72.34	60.43
FSS-TimEx	85.06	90.24	80.43	49.04	58.24

number 3818 annotated time expressions. The second dataset comprised 150 documents with 2078 timexes. This was used for evaluation and was then made available to participants, after evaluations were completed. Annotations identified the span and class of each timex. Table 3.2 shows the number of annotated timex by class in each dataset.

Table 3.2: Time expressions per dataset in the Clinical TempEval task [Bethard et al., 2015].

Class	Training	Evaluation
DATE	2583	1422
TIME	117	59
DURATION	433	200
SET	218	116
QUANTIFIER	162	109
PREPOSTEXP	305	172
Total	3818	2078

We developed a rule-based and a SVM-based approach to time expression identification that we used in Clinical TempEval. The SVM approach was developed by Dr. Genevieve Gorrell[8] (The University of Sheffield, UK) and it is described in Appendix B.

[8] http://www.dcs.shef.ac.uk/~genevieve/

3.1.1 HINX: A Rule-Based Approach

HINX is a rule-based system developed using GATE[9] [Cunningham et al., 2011]. It executes a hierarchical set of rules and scripts in an information extraction pipeline that can be split into the 3 modules: 1) text pre-processing; 2) timex identification; and 3) timex normalisation, which are described below. These modules identify and normalise temporal concepts, starting from finding basic tokens, then grouping such tokens into more complex expressions, and finally normalising their features. An additional step was included to produce the output files in the desired format.

Text Pre-processing

This module is used to pre-process the documents and identify the document creation time (DCT). We use rules written in JAPE [Cunningham et al., 2000], a GATE's pattern matching language, to identify the DCT annotation reference within the "[meta]" tag at the beginning of each document. The DCT value was split into different features to be stored at the document level – year, month, day, hour, minute, and second.

HINX also used GATE's ANNIE [Cunningham et al., 2011] – a rule-based system that was not specifically adapted to clinical domain – to provide tokenization, sentence splitting and part of speech (POS) tagging. We used the Unicode Alternate Tokenizer provided by GATE to split the text into very simple tokens such as numbers, punctuation and words. The Sentence Splitter identifies sentence boundaries, making it possible to avoid creating a timex that connects tokens from different sentences or paragraphs. POS Tagging produces a part-of-speech tag as an annotation on each word or symbol, which is useful in cases such as identifying whether the word "may" is being used as a verb or as a noun (name of the month).

Timex Identification

This module uses a set of hierarchical JAPE rules to combine 15 kinds of basic temporal tokens into more complex expressions, as described in the sequence of steps given below:

- **Numbers**: A set of rules is used to identify numbers that are written in a numeric or a non-numeric format, as numbers as words (e.g. "two and a half").
- **Temporal tokens**: Every word that can be used to identify temporal concepts is annotated as a basic temporal token, such as temporal granularities, periods of the day, names of months, days of week, names of seasons, words that represent past, present and future references, and words that can give an imprecise sense to a temporal expression (e.g. the word "few" in "the last few days"). Additionally, as a requirement for the Clinical TempEval task, we also included specific rules to identify those words that corresponded to a timex of class PREPOSTEXP (e.g. "postoperative" and "pre-surgical").
- **Basic expressions**: A set of rules identifies the basic temporal expressions, including explicit dates and times in different formats (e.g. "2014", "15th of November", "12:30"), durations (e.g. "24 hours", "the last 3 months"), quantifiers, and isolated temporal tokens that can be normalised.
- **Complex expressions**: Complex expressions are formed by connecting two basic expressions or a basic expression with a temporal token. These represent information

[9] http://gate.ac.uk

corresponding to ranges of values (e.g. “from July to August this year"), full timestamps (e.g. “Mar-03-2010 09:54:31"), referenced points in time (e.g. “last month"), and precise pre/post-operative periods (e.g. “two days postoperative").

- **SETs**: Temporal expressions denoting a SET (number of times and frequency, or just frequency) are identified by this specific set of rules (e.g. “twice-a-day", “three times every month", “99/minute", “every morning").

- **Imprecise expressions**: These kinds of expressions comprise language-specific structures used to refer to certain imprecise periods of time, including imprecise expressions defined with boundaries (e.g. “around 9-11 pm yesterday"), imprecise values for a given temporal granularity (e.g. “a few days ago", “the coming months"), precise and imprecise references (e.g. “that same month", “the end of last year", “the following days"), imprecise sets (e.g. “2 to 4 times a day"), and vague expressions (e.g. “some time earlier", “a long time ago").

Timex Normalisation

As the above identification process is run, the basic temporal tokens are combined to produce more complex annotations. Annotation features on these complex annotations are used to store specific time values, for use by the normalisation process. Such features comprise explicit values like “year=2004", references to the document creation time/DCT (e.g. “month=(DCT.month)+1" for the expression “in the following month", and “day=(DCT.day)-3" in “three days ago"), and a direct reference to the last mentioned timex in the previous sentences (e.g. “year=LAST.year" for the timex “April" in “In February 2002,... Then, in April,...").

The normalisation process uses these features to calculate corresponding final values. It also captures a set of other characteristics, including the precision of an expression, and whether or not it represents a boundary period of time. This last one is used to split the DURATION timexes (e.g. “between November/2012 and March/2013") into two different DATE expressions, as explicitly defined in the THYME Annotation Guidelines.

3.1.2 Results and Discussion

We submitted 5 runs using the HINX system and 2 runs using our SVM approach to Clinical TempEval. Results of both systems are shown in Table 3.3. For completeness, both SVM runs submitted are included. However the only difference between the two is that SVM-2 included the full training set, whereas SVM-1 included only the half reserved for testing at development time, and submitted as a backup for its quality of being a tested model. As expected, including more training data leads to a slightly superior result, and the fact that the improvement is small suggests the training set is adequate in size.

The 5 HINX runs shown in Table 3.3 correspond to the following variants: 1) using preposition “at" as part of the timex span; 2) disregarding timexes of class QUANTIFIER; 3) using full measures span for QUANTIFIERs (e.g. “20 mg"); 4) considering measure tokens as non-markable expressions; and 5) disregarding QUANTIFIERs that represent measures. The TIMEX3 type QUANTIFIER was targeted in different submitted runs as it was not clear how these expressions were annotated when comparing the training corpus to the annotation guidelines.

The HINX system got the best Recall across all Clinical TempEval systems in both subtasks. The low precision of the rule-based system was, however, a surprise, and led us to examine the training and test corpora in detail. While we would expect to see inconsistencies in any manually created corpus, we found a surprising number of repeated inconsistencies between

Table 3.3: Final Clinical TempEval results [Bethard et al., 2015].

	Span			Class		
Submission	P	R	F1	P	R	F1
Baseline: memorize	0.743	0.372	0.496	0.723	0.362	0.483
KPSCMI: run 1	0.272	0.782	0.404	0.223	0.642	0.331
KPSCMI: run 2	0.705	0.683	0.694	0.668	0.648	0.658
KPSCMI: run 3	0.693	0.706	0.699	0.657	0.669	0.663
SVM-1	0.732	0.661	0.695	0.712	0.643	0.676
SVM-2	0.741	0.655	0.695	0.723	0.640	0.679
HINX-1	0.479	0.747	0.584	0.455	0.709	0.555
HINX-2	0.494	0.770	0.602	0.470	0.733	0.573
HINX-3	0.311	0.794	0.447	0.296	**0.756**	0.425
HINX-4	0.311	**0.795**	0.447	0.296	**0.756**	0.425
HINX-5	0.411	**0.795**	0.542	0.391	**0.756**	0.516
BluLab: run 1-3	**0.797**	0.664	**0.725**	**0.778**	0.652	**0.709**

the guidelines and the corpora for certain very regular and unambiguous temporal language constructs. These included: a) timex span and class inconsistencies, b) non-markable expressions that were annotated as timexes, c) many occurrences of SET expressions that were not manually annotated in the corpus, and d) inconsistencies in the set of manually annotated QUANTIFIERs. Had these inconsistencies not been present in the gold standard, HINX would have attained a precision between 0.85 and 0.90 [Tissot et al., 2015b].

We suggest that inconsistent data such as this will tend to lower the precision of rule-based systems. To illustrate our point, we used the HeidelTime system [Strötgen et al., 2013] to produce a result on this year's dataset, and found that precision/recall were low (0.44; 0.49) despite this being a demonstrably successful system in TempEval-3. Similar low results can be observed from the ClearTK-TimeML (0.593; 0.428), used to evaluate the THYME Corpus [Styler et al., 2014]. [Styler et al., 2014] suggest that clinical narratives introduce new challenges for temporal information extraction systems, and performance degrades when moving to this domain. However, they do not consider how much the performance can be impaired by the inconsistencies found in the annotated corpus.

The appearance of a superior result by our machine learning system, which is agnostic about what information it uses to replicate the annotators' assertions, is therefore not to be taken at face value. A machine learning system may have learned regularities in an annotation style, rather than having learned to accurately find time expressions. This is an example of data bias [Hovy et al., 2014]. Machine learning systems have a flexibility and power in finding non-obvious cues to more subtle patterns, which makes them successful in linguistically complex tasks, but also gives them a deceptive appearance of success where the irregularity in a task comes not from its inherent complexity but from flaws in the dataset.

3.2 Analysis of Timexes Annotated in Clinical Notes

One way of iteratively improving annotation standards and corpora is to use human annotations to test an annotation model [Pustejovsky and Moszkowicz, 2012]. Our analysis is based on the corpus and standard that backed a recent shared annotation exercise in SemEval-

2015. The analysis of temporal expression annotation in one such corpus is an effort to gather information on the underlying model and to improve future annotation efforts.

The clarity of guidelines, skill of annotators and quality of annotated resource can be estimated by measuring agreement between annotators. Clinical TempEval's timex annotations had an IAA of 0.80 (or 0.79) [Styler et al., 2014], suggesting that these can be improved.

To investigate the quality of the dataset and annotation standard in Clinical TempEval, we have our rule-based system (HINX) based as closely as possible on the annotation guidelines, and referring to the corpus for guidance in edge cases. When evaluated using the Clinical TempEval scoring software, this system obtained good Recall (0.795 for timex spans and 0.756 for timex classes) but low precision ranging from 0.29 to 0.49. These results are low compared to the state of the art on other temporally annotated corpora.

In order to understand why our system achieved such low precision in the final Clinical TempEval results, we performed an extensive analysis of the manually annotated time expressions provided for that task, following the steps described below:

- **Manual annotations**: We tabulated all the manually annotated timexes from the Clinical TempEval corpus, listing the timex string, the timex partial sentence (including two previous and following timex tokens), the timex span (begin and end offset boundaries), and the timex class.
- **System result**: We created a similar list with the timexes identified by our system (HINX).
- **Matches & Similarities**: We compared the manual annotations with our system result to identify a) those timexes that match in terms of span and class, b) those that are similar in terms of span (at least one overlapping character), and c) those that do not have a corresponding entry.
- **Guideline reference**: For each timex that did not match, we identified the guideline, topic and section corresponding to the inconsistency.
- **Agreements & Disagreements**: We set as an "annotation agreement" each timex that a) had the exact same span and class in both manual annotated corpus and our system result, and b) complied with the annotation guidelines – an "annotation disagreement" happened when one of the previous conditions failed.
- **Found expressions**: We checked in the corpus, using a mixture of word lists and simple patterns, for additional timexes that were neither manually annotated as part of the reference corpus, nor identified by our system. We refer to the combined set of (a) manually annotated expressions, (b) expressions automatically identified by our system, and (c) these additional expressions additionally found, as the "found expressions". We will refer to this combined set of found expressions in the following subsections.

3.2.1 Annotation Analysis

We analysed the annotated datasets provided by Clinical TempEval following the methodology described in the previous Section, considering 4 types of disagreements: a) inconsistency on the annotated span and class; b) non-markable expressions; c) frequent expressions; and d) quantifiers. Each of these is explained below.

Analysis of Span and Class

When comparing the guidelines against the manually annotated corpus we can observe some inconsistencies concerning the span and the class feature of a timex. We can expect to see a degree of error in any manually annotated corpus; however, we find similar divergences occurring repeatedly. Table 3.4 summarises all the expression types we analysed, detailing the number of annotation agreements and disagreements, as well as the total number of expressions found in the corpus.

Table 3.4: Timex class and span inconsistences.

Kind of expression	Annotation Agreements	Annotation Disagreements	Found Expressions
Periods of the day	38	51	107
Temporal granularity as frequency	11	44	80
Explicit times	18	26	445
DATE modified to DURATION	35	60	95
DURATION from explicit DATEs	11	8	19
Total	113	189	746

According to TimeML Annotation Guidelines (section 2.2.3), expressions which refer to a time of the day, should be annotated as a class TIME, even if in a very indefinite way (as periods of the day, e.g. “last night" and “the morning of January 31"). From a total of 107 expressions referring to a period of the day, 89 were annotated in the corpus (more than 80%). However, we observed 51 were not annotated as a TIME, but mainly as a DATE class (less than 50% of total number of found expressions).

THYME Guidelines exemplify in section 4.2.6 that temporal granularities denoting a frequency must be annotated as a SET, for example *“monthly", “weekly", “a day", “per day", “a week", “per minute"*. However, 55% of such expressions were incorrectly annotated as DATE or QUANTIFIER (44 disagreements according to the guidelines).

Explicit times of the day should be annotated as a timex of class TIME (section 2.2.3 of TimeML guideline). This should be the case even if such expressions appear isolated in the text (e.g. *“1:33 pm"*) or within a more complex expression together with a date (e.g. *“04-Oct-2010 09:44"*). Less than 10% of the expressions denoting time were manually annotated. Of these, almost 60% represent annotation disagreements as a timex of class DATE instead of TIME.

Section 4.2.3 of THYME Guidelines state that words like “since", “during" and “until" preceding a timex of class DATE should modify the timex class to DURATION. However, in almost 65% of such modified timexes, we found that this rule was not followed, and that the timex was presented as a DATE.

Additionally, in the same section, one can find that two dates can be used to construct a DURATION timex (e.g. “December 2009 through March 2010"). However, because each one represents a single point in time, they should both be separately annotated as DATE rather than DURATION.

Non-Markable Expressions

The guidelines are clear about a diverse set of non-markable expressions. The TIDES Guidelines have a specific section (3.2) to describe what should not be annotated as a timex,

including prepositions and subordinating conjunctions, specific duration and frequency expressions, and proper names. Table 3.5 lists time expressions found in the provided corpus that are non-markable expressions according to the guidelines.

Table 3.5: Non-markable time expressions.

Expression	Annotation Disagreements	Found Expressions
Words “Date/Time"	63	359
Non-quantifiable durations	43	185
Prepositions as triggers	130	1248
Total	236	1792

There is no reference in the guidelines to annotating the words “Date" and “Time" as a timex when they are not part of a more complex expression, as such isolated words cannot be normalised. In expressions like “Date/Time=Mar 3, 2010", it is expected that “Mar 3, 2010" should be annotated as a DATE, but not the words “Date" and “Time" as time expressions of class DATE and TIME respectively. We found 359 occurrences of such words in 217 different documents, from which 63 of them were incorrectly annotated as DATE and TIME (17.5%).

Non-quantifiable durations are not markable, as they refer to some vague duration (interval) of time, including expressions like “duration", “for a long time", “some time", and “an appropriate amount of time". On the other hand, temporal expressions denoting imprecise amount of time should be annotated as a timex (e.g. *“many days", “few hours"*). We found 185 non-quantifiable duration expressions, from which 43 were incorrectly annotated as a timex with class DURATION (almost 25% of disagreement).

Prepositions which introduce noun phrases are never triggers for time expressions and they can never appear as the syntactic head of an annotated expression. In around 10% of those kind of expressions found in the corpus, time expressions were incorrectly annotated including the head preposition (*“in", “on", “at", “during", “after", “since", “until"*). Some examples include “until July", “on Monday", “in the last year".

Frequent Expressions

We observed that some expressions tend to appear more often than others in the Clinical TempEval datasets. Most of these are a timex of class SET. A SET is defined (section 4.2.6 of THYME Guidelines) as an expression which comprises a quantifier (optional) and an interval to represent a frequency (mandatory). “Three times weekly", “monthly" and “1/day" are considered as a SET, but not “twice" which is considered as a QUANTIFIER.

We selected a set of the most significant expressions, in terms of the number of occurrences, in order to compare the number of manually annotated expressions against the number of expressions which we found within the text. The expressions were organized in 7 groups:

- Present reference expressions of class DATE *“current(ly)", “recent(ly)", “now", “present(ly)"*;
- Past reference expressions of class DATE *“previous(ly)", “the past"*;

- Explicit years *“2009", “2010"*;
- Precise and imprecise expressions of class DURATION *“24-hour", “2 hours", “six-months", “years"*;
- SETs comprising number of times and frequency *“one-time daily", “two times a day", “twice-a-day", “twice-daily", “three times a day", “four times a day"*;
- SETs comprising only frequencies *“every 6 hours", “every 4 hours", “every evening", “every morning", “every bedtime"*;
- SETs following the pattern “999 /min" – such expressions are part of measurements as in *“Pulse Rate=88 /min"* or *“Resp Rate=16 /min"*.

Table 3.6 shows how many times each expression was manually annotated and how many times we found them within the corpus (number of found occurrences). Considering all of the selected expressions for this analysis, only 23.3% of such expressions were manually annotated. Considering only SET expressions, the percentage of manually annotated expression is even lower (8.5%).

Table 3.6: Frequent expressions.

Expression	Manually Annotated	Found Expressions
DATE: present reference	372	836
DATE: past reference	52	117
DATE: explicit years	55	91
DURATION: precise and imprecise	22	114
SET: times and frequency	20	1087
SET: frequency	0	216
SETs: *999 /min*	114	266
Total	635	2727

Quantifiers

A special type of timex of class QUANTIFIER was introduced in the THYME Annotation Guidelines. These are used to identify expressions such as “twice", “four times", and “three incidents" which represent the number of occurrences of an EVENT. However, the THYME Guidelines do not make it clear whether or not the words that identify the event itself should be part of the timex span.

In order to understand the way in which QUANTIFIERs and associated EVENTs should be annotated, we examined their occurrence in the Clinical TempEval corpus. We listed all non-numerical words that we found either (a) annotated as part of the QUANTIFIER span or (b) immediately after the QUANTIFIER span. Our reasoning was that these represented the repeated EVENT.

Those 20 most frequent EVENT words found in this way are detailed in Table 3.7. In the table, we compare the number of manually annotated QUANTIFIERs associated with these EVENTs in the reference corpus, with the number of all QUANTIFIERs that we could find, where

they were related to the same kind of EVENT. For example, if the reference corpus included a QUANTIFIER annotation for “twice” in the expression “twice before colonoscopy”, then we looked for all occurrences of QUANTIFIER expressions associated with “colonoscopy”. Only 11.6% of the QUANTIFIERs that we found were manually annotated in reference the corpus.

Table 3.7: Words related to quantifiers.

Related word	Manually Annotated	Found Expressions
tablet	5	1135
unit	3	117
cycle	51	65
“drinking" words*	44	53
session	4	44
pack	19	29
colonoscopy	4	27
fraction	14	22
treatment	5	16
bowel	8	16
episode	7	11
stool	7	10
beat	5	7
occasion	5	5
Total	181	1557

* “Drinking words" includes “cup", “glass", “beer", “can", “drink", “bottle", and “beverage".

Note that the THYME Annotation Guidelines explicitly exclude numeric quantifiers of objects as opposed to events, excluding for example “two units of blood". However, we included those words in our analysis as they were used as a referenced EVENT to annotate QUANTIFIERs in the corpus, usually followed by an expression which identifies frequency (e.g. “1 TABLET by mouth every evening”).

3.2.2 Recommendations

The analysis given in the previous section has led us to think about the way in which manual temporal expression annotation efforts are conducted. We venture to make a number of recommendations, hoping that these will at least be considered in future manual annotation efforts. We discuss our recommendations below.

Annotation guidelines should clearly state the full set of rules defining what should or should not be annotated, and how. For THYME, the annotators had to piece together several guidelines to figure out what to annotate. This is a potential source of error. Training in the use of multiple sets of guidelines could be considered as an alternative.

Examples are a valuable aid to annotators. Although examples are given in the THYME guidelines, the number could be expanded. In the CLEF Project for example [Roberts et al., 2009], each time an annotator raised a question, and each time persistent differences between annotators were found, new examples were added to the guidelines to re-enforce the point raised.

In creating the THYME gold standard used in the Clinical TempEval task, multiple annotators and an adjudication process were used. A potential source of error with this approach

is that where all annotators have a low recall and adjudication focuses only on resolving disputes, the resulting recall can be no greater than the union of the two. This casts doubt on the veracity of inter-annotator agreement [Fleiss et al., 1981] as an indicator of the accuracy of annotation of a corpus.

Some constructs and guidelines can be represented by simple, unambiguous rules, and where this is the case, the rules will most likely outperform the human annotator in terms of recall. This last point raises the potential merit of using a rule-based system to prepare a corpus, creating annotations for review by human annotators. We feel that in such high recall cases, the disadvantage of the approach, that there tends to be a poor correction of missing spans, would be outweighed by the increased number of annotations found.

3.3 Imprecise Temporal Data in Text

Considerable effort has been put into the extraction of temporal information from natural language texts, allowing systems to deal with complex temporal questions. However, the temporal intervals expressed in natural language are often vague, making it necessary to extend traditional temporal reasoning formalisms to cope with the vagueness [Schockaert et al., 2008].

Imprecise timexes cloud later temporal processing. For example, they make it hard to evaluate whether an event should be included in a query result that involves timeline evaluation. In the sentence “frequent headaches for less than one month", a patient tries to describe how long a headache has lasted. The corresponding amount of time, however, cannot be accurately defined, due to the modifier “less than". The target imprecise expression “less than one month" is annotated in TimeML as:

```
<TIMEX3 value="P1M" mod="LESS_THAN">
less than one month
</TIMEX3>
```

As a consequence, when interpreting this expression and its annotated features, it is not clear whether we should consider each possible number of days between 0 and 30 as equally likely, or whether for example, 20-25 days ago is more likely than 5-10 days ago or even “yesterday".

3.3.1 Quantifying imprecise timexes

In order to understand the relevance of normalising imprecise temporal information in different domains, we analysed a set of three clinical and six non-clinical corpora in English and Portuguese (Table 3.8) to compare the occurrence of imprecise timexes in both general and specific domain data. We used the HINX system [Tissot et al., 2015a] to identify the occurrence of imprecise timexes. HINX asserts a specific annotation feature ($precision$ = “$imprecise$”) to identify imprecise timexes, based on a set of rules to identify words, expressions and specific language structures that represent imprecision.

Table 3.9 compares the number of imprecise temporal expressions against the total number of timexes in each corpus, and shows that imprecise timexes in clinical corpora can comprise up to 35% (SLAM corpus) of the temporal expressions. The percentage of imprecise expressions found in newswire was no more than 13% (WikiWars corpus).

Table 3.10 describes the distribution of imprecise timexes in terms of temporal granularity. The temporal granularity is the time granularity used to compose the timex, as DAY in “in less than 15 days", or UNDEFINED in “more recently". The set of expressions with granularity

Table 3.8: Corpora analysed about the occurrence of precise and imprecise timexes.

Corpus	Language	Documents	Description
AQUAINT	English	73	News reports, also referred to as the Opinion Corpus, annotated with time expressions [Pustejovsky et al., 2010].
TE3 Platinum	English	20	The corpus used to rank participant systems in the TempEval-3 evaluation exercise, consisting of newswire documents and blog posts annotated for events, temporal expressions and temporal relations [UzZaman et al., 2013a].
TE3 Silver	English	2,452	Documents automatically annotated as a silver standard in TempEval-3 [UzZaman et al., 2013a].
TimeBank	English	183	News articles annotated with temporal information, events, times and temporal links between events and times [Pustejovsky et al., 2003b].
WikiWars	English	22	Documents sourced from Wikipedia, within the domain of military conflicts, containing temporal expressions annotated with TIMEX2 [Mazur and Dale, 2010].
CSTNews4	Portuguese	50	A discourse-annotated corpus for fostering research on single and multi-document summarization from news texts [Cardoso et al., 2011].
* THYME	English	248	Clinical narratives used as training and evaluation datasets in SemEval-2015 Clinical TempEval Task [Bethard et al., 2015].
* SLAM	English	1,000	Medical records without any pre annotated timexes provided by the Biomedical Research Centre and Dementia Biomedical Research Unit at South London and Maudsley NHS Foundation Trust and King's College London [Stewart et al., 2009].
* InfoSaude	Portuguese	3,360	Medical records without any pre annotated timex extracted from the *InfoSaude* system, Public Health Department in Florianopolis, Brazil [Bona, 2002].

* Clinical corpora

YEAR, MONTH, WEEK and DAY represents more than 60% of the total amount of imprecise expressions in both clinical and non-clinical corpora. Imprecise expressions denoting time (HOUR, MINUTE, and SECOND) represent less than 5% of imprecise expressions in non-clinical data and less than 3% in clinical corpora.

Finally, Table 3.11 shows the distributions of imprecise temporal expressions found in clinical corpora according to each of the main temporal classes defined by TimeML (DATE, TIME, DURATION, and SET). The occurrence of imprecise timexes is concentrated on the classes DATE and DURATION for clinical documents. In non-clinical documents, the occurrence of imprecise timexes is concentrated on the class DURATION.

3.3.2 Classification of Imprecise Timexes

We analysed the full set of imprecise expressions found in clinical corpora in order to understand the different ways the imprecision can be expressed in natural language. We defined 6 main groups of imprecise timexes according to their main language elements:

Table 3.9: Occurrence of Imprecise Timexes.

Non-clinical data				Clinical data			
	Total number of Timexes	Imprecise Timexes	Imprecise %		Total number of Timexes	Imprecise Timexes	Imprecise %
AQUAINT	463	35	7.6%	Thyme	3,358	659	19.6%
TE3 Platinum	158	20	12.7%	SLAM	35,120	12,226	34.8%
TE3 Silver	15,191	863	5.7%	InfoSaude	503,005	53,830	10.7%
TimeBank	478	60	12.6%	General	134,388	13,785	10.3%
WikiWars	862	112	13.0%	Gynecology	66,021	5,452	8.3%
CSTNews4	444	32	7.2%	Nutrition	64,282	6,286	9.8%
				Psychiatry	238,314	28,307	1.98%
Total (micro)	17,596	1,122	6.4%	Total (micro)	541,483	66,715	12.3%
Total (macro)			9.8%	Total (macro)			21.7%

Table 3.10: Occurrence of imprecise timexes by granularity.

Imprecise Granularity	Non-Clinical Corpora	Clinical Corpora
Year	28.5%	21.1%
Month	20.1%	21.2%
Week	7.7%	6.8%
Day	10.7%	17.6%
Time (Hour, Minute and Second)	4.9%	2.8%
Undefined	23.8%	15.2%
Others*	4.3%	15.3%

*"Others" includes Century, Decade, Quarter and Season.

1. **Present Reference (PR)**: a time reference related to the present, based on the document creation time (DCT) (e.g. "now", "recently", "currently");

2. **Modified Value (MV)**: an imprecise timex comprising a modified precise amount of time (e.g. "approximately 10 days", "less than a month");

3. **Imprecise Value (IV)**: an expression built around a certain imprecise amount of time (e.g. "some days", "several weeks"), or formed with undetermined amount of time, in which granularity is usually presented in the plural, with the absence of numeric values (e.g. "years");

4. **Range of Values (RV)**: an amount of time defined by boundaries (e.g. "every 3-4 months", "between 8-10 years");

5. **Partial Period (PP)**: a portion of time within a larger time frame (e.g. "the end of last year", "middle of January");

6. **Generic Expression (GE)**: an expression denoting a generic period or amount of time (e.g. "this time", "at the same time").

Table 3.12 details the number of imprecise timexes found in each clinical corpus according to the imprecise group.

Table 3.11: Imprecise Timexes by Class in clinical corpora.

Corpus	DATE Tot	DATE Imp	DATE %	TIME Tot	TIME Imp	TIME %	DURATION Tot	DURATION Imp	DURATION %	SET Tot	SET Imp	SET %
THYME	2,588	460	17.8%	118	13	11.0%	434	150	34.6%	218	36	16.5%
SLAM	22,678	9,296	41.0%	919	27	2.9%	8,001	2,801	35.0%	1,558	102	6.5%
SMS	210,596	19,082	9.1%	63,468	71	0.1%	190,411	34,524	18.1%	38,530	153	0.4%
Avg (micro)	235,862	28,838	12.2%	64,505	111	0.2%	198,846	37,475	18.8%	40,306	291	0.7%
Avg (macro)			22.6%			4.7%			29.2%			7.8%

Table 3.12: Timexes by Imprecise Type in clinical corpora.

	Clinical Corpora		
Imprecise Type	THYME	SLAM	InfoSaude
PR	55.7%	58.0%	30.2%
MV	15.5%	6.6%	27.0%
IV	11.9%	14.4%	24.9%
RV	10.2%	4.0%	13.6%
PP	6.2%	3.2%	4.3%
GE	0.5%	13.8%	0.0%

We chose to apply and test the proposed methodology starting by the three most representative kinds of imprecise expressions in terms of occurrence (PR, MV, and IV). The PR imprecise type represents more than 50% of imprecise timexes in the clinical domain. However, it comprises expressions devoid of a temporal granularity, requiring distinct questionnaire design and input data representation.

3.4 Normalisation of Imprecise Timexes

Normalisation of an imprecise temporal expression depends on how people reason about imprecise information. Reasoning about an imprecise timex in a specific context, such as in clinical text, may depend on a broader narrative analysis, and an understanding of the context in which the expression was created. Consider, for example, the following expression: “symptom Y observed for less than a month". The underlined imprecise timex could mean a period near to, but less than 30 days. However, in “drug X is only recommended for symptom Y when it has been present for less than a month", the same expression imposes a time limit restriction and the literal interpretation of a 0-30-day period, in turn, seems appropriate.

Despite this possible influence of different contexts on the interpretation of imprecise timexes, we present a methodology on how to produce normalisation models for each different imprecision type according to the people’s common cognitive perception of temporal imprecision.

3.4.1 Specification of the Input Data

In order to collect data on how people interpret vague descriptions of time in text, we designed two questionnaires – in Portuguese[10] and English[11] – each question aims to capture the perception about an imprecise value for a given imprecise timex.

[10] http://staffwww.dcs.shef.ac.uk/people/H.Tissot/quiz/Portuguese/

[11] http://staffwww.dcs.shef.ac.uk/people/H.Tissot/quiz/English/

Each question shows a sentence comprising 2 to 3 descriptions of time that could be precise or imprecise. The target imprecise timex to be evaluated is underlined. The Portuguese questionnaire comprises 125 questions split into 5 questionnaires (25 questions each), each question made with modified sentences found in a set of medical records from the *InfoSaude* corpus. The English version has a total of 150 questions split into 10 questionnaires (15 questions each), each question designed using fictional text to capture the perception about specific imprecise value for a given set of imprecise timexes (non-clinical). The types of questions covered by each questionnaire are described in Table 3.13. See Appendixes C and D for a list of sentences used to design each questionnaire.

Table 3.13: Types of questions in each questionnaire.

Imprecise Type	Question Type	Number of questions	
		Portuguese	English
MV	Approximately	30	38
	Less Than	18	26
	More Than	24	26
IV	Imprecise Value	41	48
PR	Present Reference	12	12
	Total	125	150

MV and IV questions in the Portuguese survey asked for a specific number of days, weeks, months, or years (e.g. for “more than 10 days", one specific number of days should be selected, with options ranging from 7-60 days). The same type of question in English asked for a possible range of time (e.g. for “more than 5 days", a range of days start-end should be selected, with start point ranging from 0-40 days and end point ranging from 0-60 days). An additional option “more than 60 days" was also included on the questions covering the MV imprecise type. TR questions (“now", “currently", “recently") asked for a temporal granularity that would better describe when the associated event starts. We wanted to test different ways to answer each question, leading to the mentioned differences in the design of each questionnaire in terms of how the answers should be entered.

As most of the imprecise temporal expressions found in the documents we had previously analysed refer to the classes DATE and DURATION, we considered “1 day" as being the basic and minimal unit of time in the experiments. We used a discrete set of an integer number of days, disregarding granularities having TimeML TIMEX3 type TIME (hours, minutes and seconds).

The Portuguese survey was approved by the *InfoSaude* Research Committee and submitted to 50 universities in Brazil, covering students and staff member from different departments, from which we gathered a total of 352 submissions. The English survey was approved by the University of Sheffield’s Research Ethics Committee and submitted to all student and staff members of an opt-out mailing list in that institution. We gathered a total of 890 submissions in English.

3.4.2 Membership Functions

We aim to normalise imprecise expressions through the use of fuzzy membership functions (MSF). The membership function would place an imprecise timex in the timeline with a

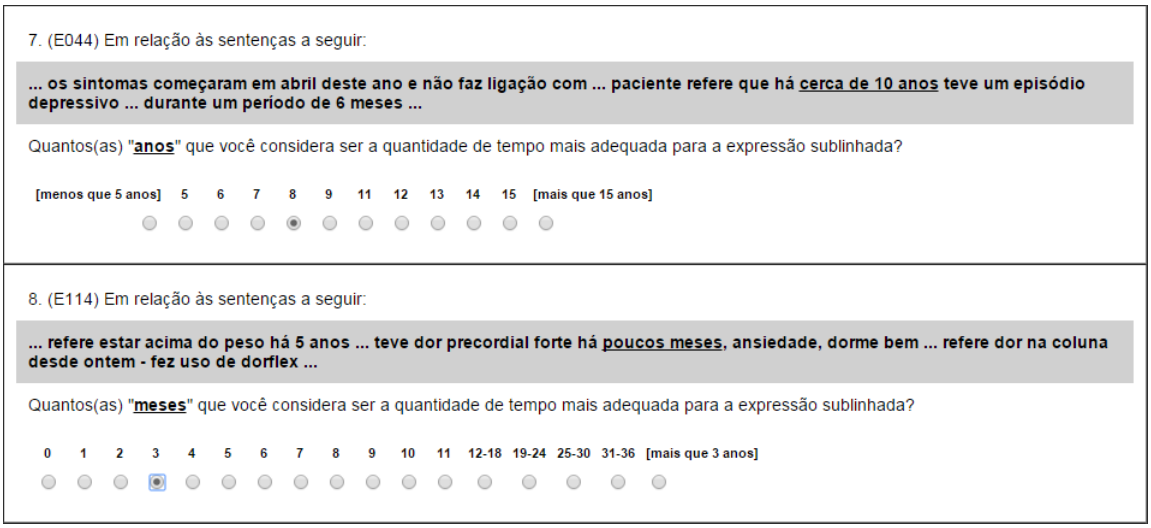


Figure 3.1: Example of questions used to design the questionnaire in Portuguese.

Figure 3.2: Example of questions used to design the questionnaire in English.

certain confidence level. In addition, a search result would have additional information indicating the confidence score for each event associated to an imprecise timex.

A fuzzy set is the basic concept that underlies fuzzy systems theory [Pedrycz and Gomide, 1998]. A fuzzy set allows us to capture, represent, and work with linguistic notions of impreciseness, unpredictability, and vagueness. A fuzzy set S is characterized by a membership function M mapping the elements of a (finite or not) domain, space or universe of discourse T into the unit interval $[0, 1]$ [Zadeh, 1994].

A membership function M can be defined in different forms, such as triangular or trapezoidal functions, or continuously differentiable curves with smooth transitions, such as normalised Gaussian functions. The *height* of a fuzzy set S is the largest membership grade of any element in that set, whereas a fuzzy set S is called *normal* when $height(S) = 1$, and *subnormal* otherwise [Pedrycz and Gomide, 1998].

Given a list of membership functions for the same kind of imprecise expression (e.g. of the form “less than N days"), we want to produce a generic model where, given N as an input, the model can calculate the parameters to describe a membership function for all expressions of that type.

We used two types of membership functions: trapezoidal (4-point-based) and hexagonal (6-point-based) membership functions. A trapezoidal membership function is defined by a set of 4 parameters $M_4(p, r, s, v)$, such as $p < r <= s < v$, p and v are the boundary limits where the confidence is 0, r and s are the boundary limits where the confidence is 1. When $r = s$, the MSF shapes like a triangular function. A hexagonal (6-point-based) membership function is defined by a set of 6 parameters $M_6(p, q, r, s, t, v)$, such as $p < q < r <= s < t < v$, and additionally the the trapezoidal boundaries, q and t are the values where the confidence is 0.5.

Trapezoidal and hexagonal membership functions were chosen because: a) they are asymmetrical and can have their shapes adapted flexibly to match different patterns, and b) their linear boundaries make them easier to use in terms of computing fuzzy logical and relational operations.

3.4.3 Pre-processing

For each question within the questionnaires we performed the following steps to pre-process the collected data to approximate the corresponding M_4 and M_6 membership functions:

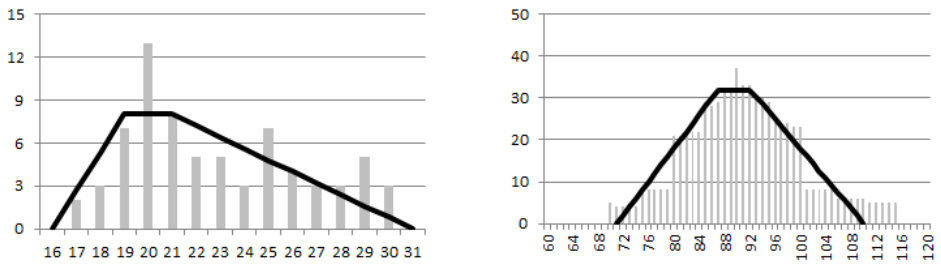
1. We split the total set of answers into two datasets (50:50%) to be used as training and testing datasets.

2. For each question we calculated a histogram based on the number of answers given to each possible option.

3. Each histogram was approximated to a trapezoidal and to a hexagonal membership function, using a full search method in order to minimise the approximation error.

We looked for the best combination of values for the parameters $M_4(p, r, s, v)$ or $M_6(p, q, r, s, t, v)$, and the best membership function height in the y axis, which corresponds to the number of given answers.

Figure 3.3(a) shows the histogram and trapezoidal function obtained for the expression “less than 30 days" from the survey in Portuguese, defined as $MSF_{LessThenP30D}(16, 19, 21, 31)$ – parameters (p, r, s, v) represent number of days, and the confidence = 1 at the height = 8 in the histogram. Similarly, Figure 3.3(b) presents the histogram and approximated trapezoidal function for the expression “about 3 months" from the questionnaire in English, defined as $MSF_{ApproxP3M}(71, 87, 92, 110)$ – the confidence = 1 at the height = 32 in the histogram.



(a) “less than 30 days" (Pt) (b) “about 3 months" (En)

Figure 3.3: Histogram and trapezoidal function for two imprecise timexes.

3.4.4 A Methodology to Normalise Imprecise Timexes

We compared different approaches, such as linear regression, and machine learning, to model each kind of imprecise expression. In order to identify which method best models each type of imprecise timex, we explored a diverse set of alternatives. The following steps were performed to analyse the data collected from the questionnaire described in Section 3.4.1:

1. Initially, input data collected from the questionnaire was pre-processed. For every question we calculated the distribution of answers in the form of a histogram. A trapezoidal and a hexagonal membership functions were approximated to describe the given histogram.

2. For those questions using temporal granularity other than “DAY" we attempted to use both options when training the models, (a) the original granularity and the numeric value (Val) extracted from the temporal expression as it was with its original granularity (e.g. “3" in “about 3 months"), and (b) the same expression converted to the granularity of days (Day) (e.g. “3" in “about 3 months" was converted to “90 days").

$$
\begin{aligned}
LessThan(n) &= [0.7 * n, 1.0 * n] \\
Approx(n) &= [0.8 * n, 1.2 * n] \\
MoreThan(n) &= [1.0 * n, 1.3 * n] \\
FEW &= [2, 3] \\
SOME &= [4, 5] \\
MANY &= [6, 8] \\
SEVERAL &= [9, 12] \\
Undefined &= [8, 20]
\end{aligned}
$$

Figure 3.4: Unsupervised baseline parameters for IV and MV expressions.

3. For each expression type, we defined range-based unsupervised parameters to use as baseline, which were arbitrary, manually chosen. Figure 3.4 shows the unsupervised interval parameters defined for MV and IV questions.

4. In order to produce a generic model that could be used to calculate any membership function for a given imprecise timex type, we applied four different variations of a linear regression to generalise each one of the parameters used to define a trapezoidal (p, r, s, v) or hexagonal (p, q, r, s, t, v) membership function for each given type of imprecise timex: a) the usual ($y = a + b * x$) linear regression (Lin-A); b) we forced the independent constant a in the linear formula to be equals to zero (Lin-0); c) the linear regression with the natural logarithm values of each expression ($ln(y) = a + b * ln(x)$), in an attempt to map those expression given in terms of years (e.g. “5 years" = “1825 days") as close to those describing periods of days or weeks (Log-A); and lastly, d) the linear regression based on the logarithm values was extended to force $a = 0$ (Log-0).

5. For those timexes comprising imprecise values (IV), we also calculated the mean (MEAN) values of each membership function parameter, combining the normalised values described in 2 (Val and Day) and 4 (Lin and Log).

6. For those timexes comprising imprecise values (IV) and present references (PR), the linear regression and machine learning variations used the temporal context as input value. This approach was used in an attempt to evaluate whether the perception of a present reference imprecise timex would be influenced by the temporal context distance.

   Definition 3.1 (Temporal Context) *Temporal Context is the distance in days between the current date (DCT - document creation time) and the last timex mentioned in the sentence prior to the imprecise timex being evaluated. For the designed questionnaires, DCT was defined as the date when each questionnaire was published.*

7. For MV and IV types of imprecise expression, we used a multilayer perceptron (MLP) with the Backpropagation algorithm [Gardner and Dorling, 1998] to learn how to return the membership function parameters for a given imprecise timex. We also combined the normalised values described in 2 (Val and Day) and 4 (Lin and Log). We used k-fold cross-validation to select the best model with $k = 4$. The internal MLP structure and learning parameters were chosen in a training setup step, after testing and comparing different configuration settings. Table 3.14 describes features and parameters used in the training step. In order to test the hypothesis that Present Reference (PR) expressions understanding could be influenced by the temporal context, we only tested the linear regression approach for that kind of expressions.

Table 3.14: MLP parameters and features used.

Type	Name	Description (Value)
Features	Granularity	Four input values to set the temporal granularity – "Val" variation
	Reference Value	Number extracted for MV expressions – "Val" variation
	Reference Days	Number of days extracted for MV expressions – "Day" variation
	Temporal Context	Number of days that represents the temporal context – IV expressions
	Imprecise Value	Five input values to set the imprecise value – IV expressions
Training Parameters	maxIteration	Maximum number of training iterations to be performed (5000)
	minIteration	Minimal number of iterations to be performed before stopping (1000)
	maxNoBetter	Training stops after 200 iterations with no improvement (200)
	K	Number of folds in K-Fold Cross Validation (4)
MLP Design	hiddenLayer	Number of neurons in the hidden layer ($(inputLayerSize - 1) * (outputLayerSize - 1)$)
	outputLayer	Number of neurons in the hidden layer to produce trapezoidal MSFs (4) or trapezoidal MSFs (6)
	learningRate	Learning rate used by the backpropagation algorithm (0.95)

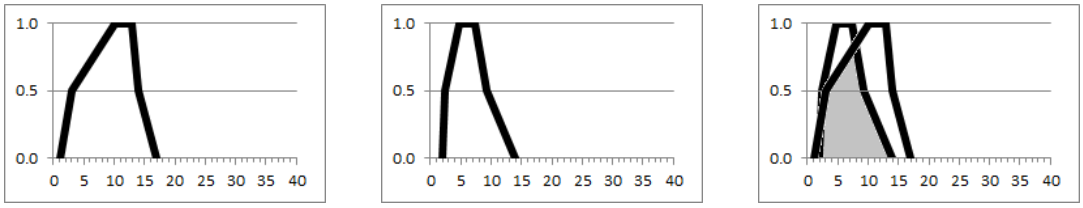
8. In order to evaluate each model we compared each individual membership function generated by the given model with the equivalent membership functions from the testing dataset. We used the areas of each membership functions to produce the F1-score, which defines how much the two functions areas overlap. Partial areas that do not overlap are considered false positive and false negative areas, and the overlap is considered as a true positive area. When $F1 = 1$ both membership functions are exactly the same, and when $F1 = 0$ there is no overlap between those given functions. The F1-score for the entire model was calculated using the average F1-score from all the membership functions used to test the model.

9. Finally, for each type of imprecise timex, we used the average F1-score obtained from all the different expression variations and between the trapezoidal and hexagonal membership functions in order to compare and select the most appropriate normalisation model. Figure 3.5 shows two hexagonal membership functions – A(1,3,10,13,14,17) and B(2,3,5,7,9,14) – and the visual representation of the F1-score between A and B, meaning the percentage of the common area relative to the total area of both functions. In the illustrated example, F1-score resulted 0.6567.

10. Alternatively, we compared models created for imprecise temporal expressions in different languages (English and Portuguese). We calculated the F1-score between both languages as the average of each F1-score calculated for each expression format. However, when calculating the F1-score using the MSF area, it was not possible to identify whether the differences are more concentrated in the top (confidence=1) or the bottom (confidence=0) of such functions. In order to try to identify where such differences are concentrated, we used a variation of F1-score that we called $F1_{3D}$. For each MSF we considered a third dimension that identifies how deep each MSF is, varying from 0 at the bottom to 1 at the top. Instead of using the MSF areas, we then used the MSF volumes to calculate $F1_{3D}$. Figure 3.6 illustrates the difference between F1 and $F1_{3D}$, comparing three MSFs (A, B, and C). A and B have a difference in the top, whilst A and C have the exact same difference in terms of area, in the bottom instead. Thus, $F1(A,B) = F1(A,C) = 0.9655$. When calculating the $F1_{3D}$, we can observe $F1_{3D}(A,B) < F1_{3D}(A,C)$, what means A and B have differences concentrated more in the top comparatively to the differences between A and C – differences at the top have more influence to decrease $F1_{3D}$ than differences at the bottom.

$$F1(A,B) = \frac{2 \times CommonArea(A,B)}{Area(A) + Area(B)} \tag{3.4}$$

$$F1_{3D}(A,B) = \frac{2 \times CommonVolume(A,B)}{Volume(A) + Volume(B)} \tag{3.5}$$



(a) Hexagonal MSF A (b) Hexagonal MSF B (c) F1 score representation

Figure 3.5: F1-score representation between membership functions A and B.

3.5 Results

In this section we present the results of the analysis for the evaluated imprecise types (MV, IV, and PR). In order to represent normalisation models, we developed a graphical representation where we plot both the training data, and the produced generalisation model. Figure 3.7 shows how that graphical representation works. Each known membership function produced from

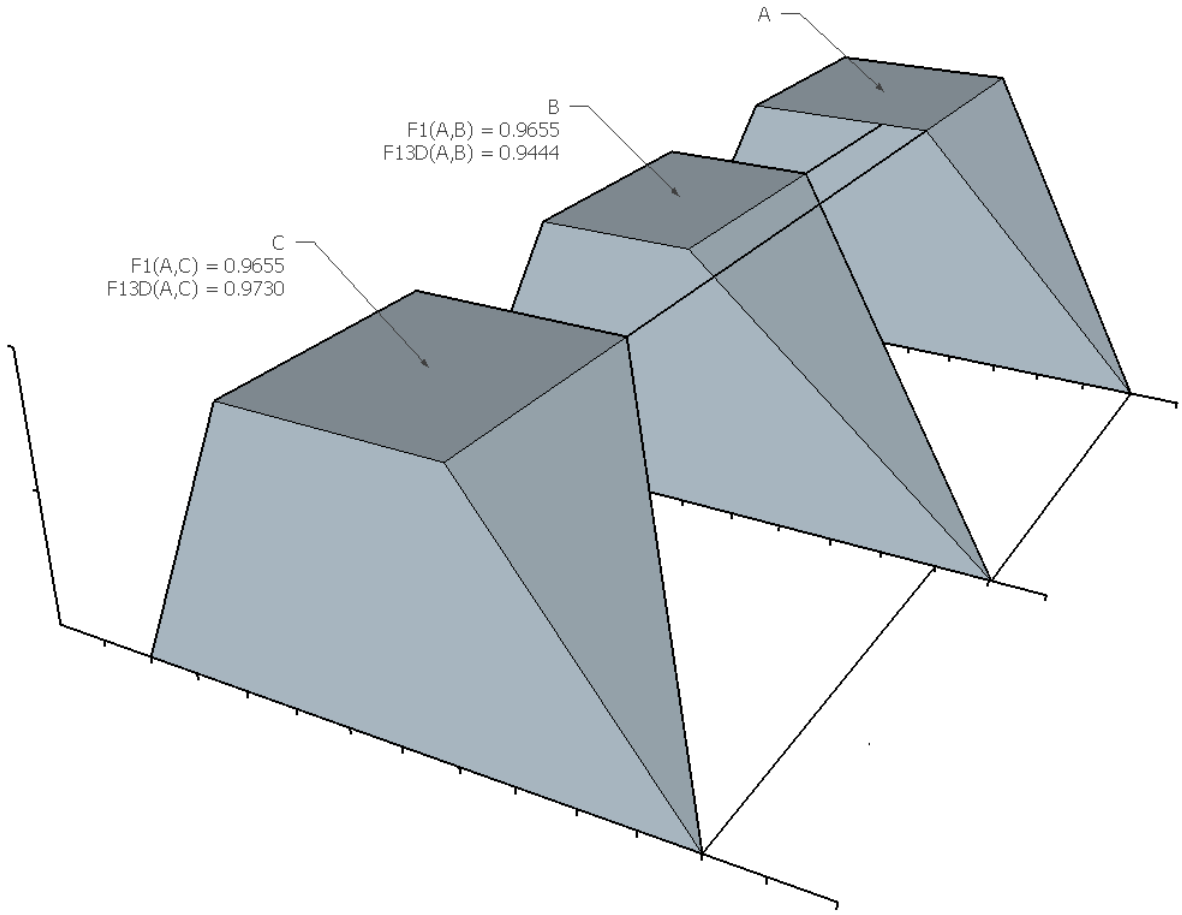


Figure 3.6: Different F1 and $F1_{3D}$ scores used to calculate the similarity between membership functions.

the input data is plotted as a vertical bar, with a dark central area representing the top of the MSF, where confidence is 1. The bottom and the top of each vertical bar represent the MSF limits where confidence is 0. The grey area in the background is the generalization for a given expression type, as when we need to normalise an unknown expression, the normalisation model will give us the parameters that describe the corresponding MSF for that expression. For example, the selected area in the right side represents the limits for an unknown expression “less than 90 days", which would be defined as $MSF_{LessThanP90D}(23, 65, 85, 96)$.

3.5.1 Modified Value (MV) Expressions

Table 3.15 compares the results of each model used to produce trapezoidal (M_4) and hexagonal (M_6) membership functions for the group of expressions comprising “less than", “more than", and “approximately" subtypes for both languages (Portuguese and English). Different models can be compared using the average (Avg) score between M_4 and M_6.

The Log-A variation achieved the best score for this kind of expression among all the Linear Regression variations for both Languages. The MLP approach produced a result that is better than the Log-A variation in Portuguese. However, MLP achieved a result that is similar to the baseline in English.

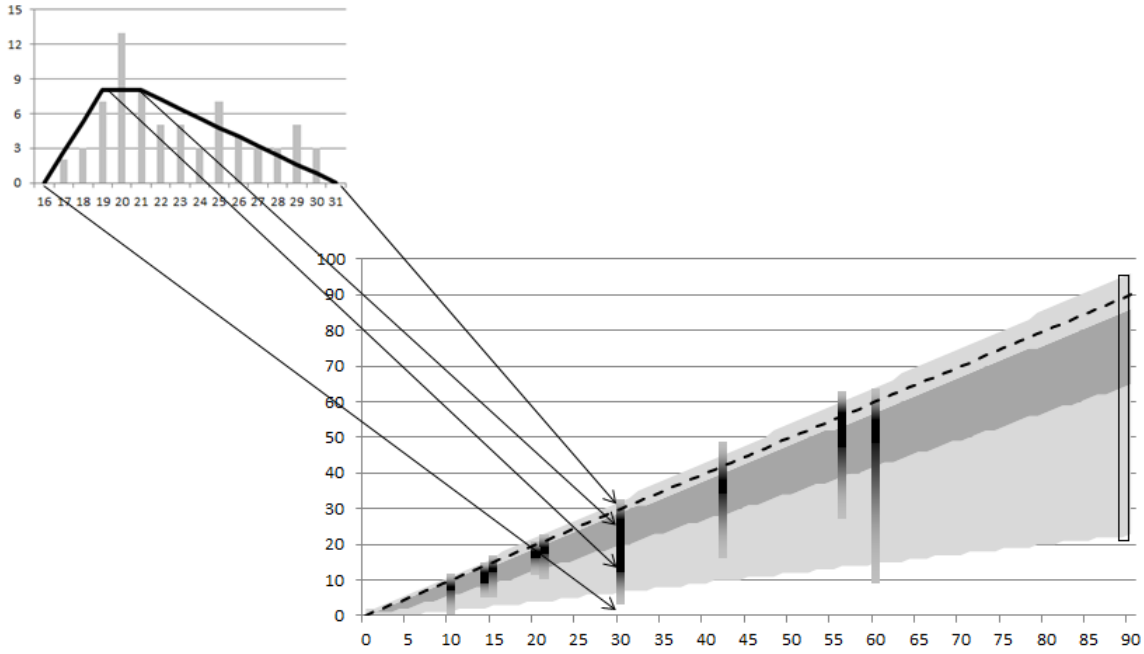


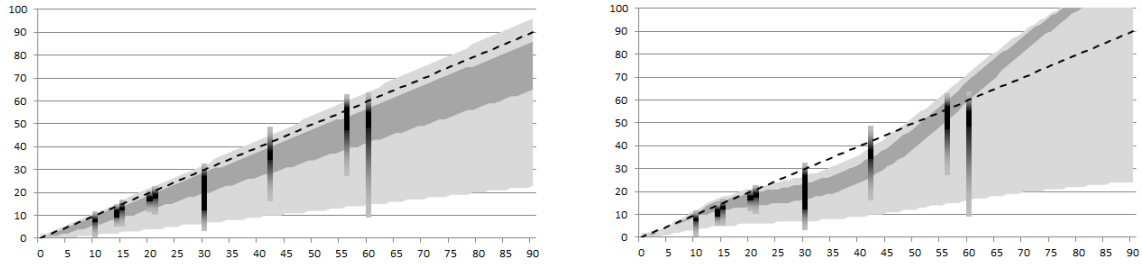
Figure 3.7: Graphical representation of a generalisation model.

Table 3.15: F1-scores for MV temporal expressions in Portuguese and English.

Method	Var	Portuguese			English		
		M_4	M_6	Avg	M_4	M_6	Avg
Baseline		0.6734	0.6465	0.6600	0.7411	0.7310	0.7360
Regression	Lin(A)	0.6357	0.5589	0.5973	0.6158	0.6134	0.6146
Regression	Lin(0)	0.7621	0.7406	0.7513	0.7976	0.7948	0.7962
Regression	Log(A)	0.7723	0.7469	**0.7596**	0.8147	0.8062	**0.8105**
Regression	Log(0)	0.6698	0.6614	0.6656	0.6789	0.6938	0.6864
MLP	Day/Lin	0.3214	0.5842	0.4528	0.3400	0.5140	0.4270
MLP	Day/Log	0.7297	0.7559	0.7428	0.6790	0.7869	0.7330
MLP	Val/Lin	0.7851	0.7421	**0.7636**	0.7388	0.7878	0.7633
MLP	Val/Log	0.7572	0.7380	0.7476	0.7609	0.7742	**0.7675**

Figure 3.8(a) shows the model using the Log-A Linear Regression variation, and Figure 3.8(b) shows the model from MLP-Val/Log, both used to produce trapezoidal functions for expressions of the form “less than N days" in English. The grey area in the chart corresponds to the generalised model, and each vertical bar represents a membership function from the testing set. The darker area corresponds to the interval delimited by the parameters r and s (confidence = 1 in each membership functions). The MLP model is consistent when producing membership function parameters that are inside the limit boundaries used to train the given model. However, it is not consistent when trying to produce membership function parameters that are outside those limits. For instance, it finds values for the parameters r and s that are greater than N for “less than N days" for each $N > 60$ (darker grey area in the chart). Similar differences between Linear Regression and MLP approaches were observed in the Portuguese models. Linear Regression models are more consistent when generalising MV imprecise timexes.

Comparing MV imprecise expressions, we found $F1 = 0.731$ and $F1_{3D} = 0.695$ as the similarity between the Log(A) regression models resulted for Portuguese and English.



(a) Linear Regression (Log-A)

(b) MLP (Val/Log)

Figure 3.8: Generalisation of “less than X days" expressions within the period of 0-90 for two different approaches in English.

3.5.2 Imprecise Value (IV) Expressions

Table 3.16 compares the results of each model used to produce trapezoidal and hexagonal membership functions for the IV type of temporal expressions. Linear Regression and MLP methods used the distance (in days) to the last precise temporal expression found in the text prior to the target imprecise timex as an input parameter when creating each model. We used two MLP approaches: a) one to learn each temporal granularity (“days", “weeks", “months", “years"), and b) one to learn each imprecise value (“few", “some", “many", “several").

Table 3.16: F1-scores for IV temporal expressions in Portuguese and English.

		Portuguese			English		
Method	Var	M_4	M_6	Avg	M_4	M_6	Avg
Baseline		0.3251	0.3116	0.3184	0.3184	0.2987	0.3086
Mean	Day/Lin	0.6616	0.8477	0.75466	0.8667	0.8478	**0.8573**
	Day/Log	0.6574	0.8488	0.75309	0.8671	0.8462	0.8567
	Val/Lin	0.6692	0.8509	**0.76005**	0.8591	0.8454	0.8522
	Val/Log	0.6562	0.8476	0.75192	0.8440	0.8314	0.8377
Regression	Day/Lin	0.6608	0.8508	0.75580	0.8922	0.8716	0.8819
	Day/Log	0.6606	0.8461	0.75334	0.8840	0.8683	0.8761
	Val/Lin	0.6735	0.8582	**0.76586**	0.8892	0.8779	**0.8835**
	Val/Log	0.6685	0.8417	0.75510	0.8473	0.8487	0.8480
MLP	Day/Lin	0.6109	0.7792	0.69507	0.7924	0.8272	0.8098
(Granularity)	Day/Log	0.6944	0.7286	0.71153	0.8491	0.8143	0.8317
	Val/Lin	0.8202	0.7673	**0.79377**	0.8480	0.8321	0.8401
	Val/Log	0.7514	0.7264	0.73892	0.8483	0.8197	**0.8340**
MLP	Day/Lin	0.6262	0.5821	0.60418	0.7609	0.7578	0.7593
(Imprecise Value)	Day/Log	0.7127	0.5517	0.63219	0.8624	0.8438	**0.8531**
	Val/Lin	0.7842	0.7385	0.76135	0.8210	0.8115	0.8163
	Val/Log	0.7623	0.7663	**0.76429**	0.8418	0.7623	0.8020

The best average F1-scores for each evaluated method are similar (ranging from 0.76 to 0.79 in Portuguese, and from 0.84 to 0.88 in English). The best average F1-score was achieved by the MLP model trained based on Granularities in Portuguese and by the Linear Regression (Val/Lin) in English. However, the Mean method has a similar average score and is the only method which was created independently of an input.

Figures 3.9 and 3.10 show the hexagonal functions created by the method Mean(Day/Lin) for the IV timexes in Portuguese and English. We calculated the F1-score that represents the similarity between both languages. Each expression comprising the same type were compared to

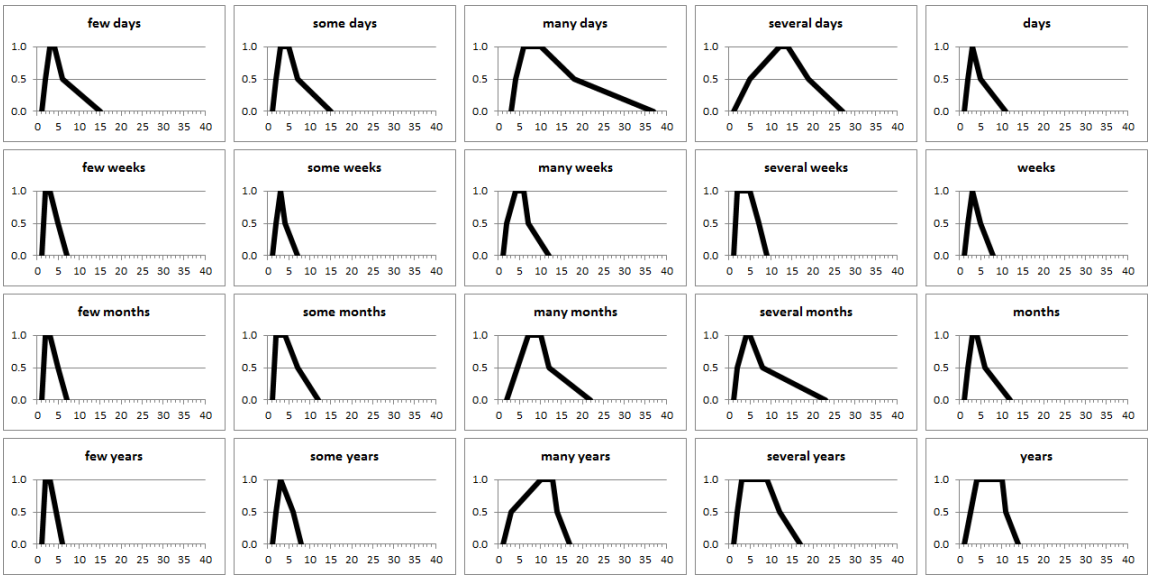


Figure 3.9: Hexagonal membership functions for IV imprecise timexes in Portuguese.

calculate a partial F1-score (e.g. “some days" in Portuguese and the same expression in English) as the average between F1-score for the trapezoidal and hexagonal MSFs. The calculated average F1 score among all the IV expressions resulted 0.767 as the similarity between Portuguese and English. Similarly, the $F1_{3D}$-score between both languages resulted 0.719, indicating there are more differences between languages at the top of the MSFs than at the bottom.

Table 3.17 depicts the pairs (F1,$F1_{3D}$) for each IV expression format, the averages for each temporal granularity and each imprecise value, and the global average considering all IV expression (in bold).

Table 3.17: F1 and $F1_{3D}$ scores comparing Portuguese and English for IV expressions.

	Days	Weeks	Months	Years	Average
Few	(0.784 , 0.797)	(0.785 , 0.729)	(0.907 , 0.846)	(0.767 , 0.714)	(0.811 , 0.772)
Some	(0.617 , 0.599)	(0.895 , 0.840)	(0.873 , 0.871)	(0.700 , 0.676)	(0.771 , 0.747)
Many	(0.866 , 0.833)	(0.807 , 0.799)	(0.873 , 0.871)	(0.598 , 0.431)	(0.786 , 0.734)
Several	(0.316 , 0.191)	(0.832 , 0.782)	(0.794 , 0.837)	(0.788 , 0.699)	(0.683 , 0.627)
Undefined	(0.724 , 0.640)	(0.690 , 0.615)	(0.904 , 0.867)	(0.829 , 0.734)	(0.787 , 0.714)
Average	(0.661 , 0.612)	(0.802 , 0.753)	(0.870 , 0.858)	(0.737 , 0.651)	**(0.767 , 0.719)**

3.5.3 Present Reference (PR) Expressions

Present Reference (PR) imprecise timexes comprise those expressions including “currently", “recently", and “now". For this kind of imprecise timexes we asked people to choose the most appropriate option to express the amount of time since when the event associated with the target expression occurred. Figure 3.11 shows two examples of questions extracted from each questionnaires in English and Portuguese. In each question, the target imprecise expression should be defined by another imprecise timex. Options included 4 IV expressions: “days", “weeks", “months", and “years".

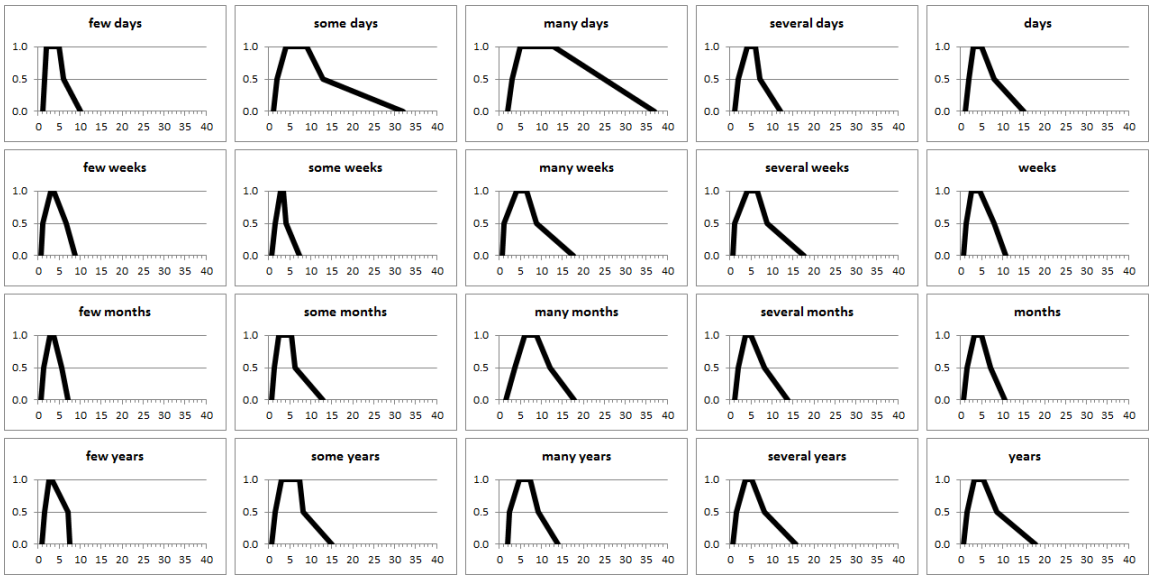


Figure 3.10: Hexagonal membership functions for IV imprecise timexes in English .

According to Figure 3.10 each IV expression is defined with the following parameters (in number of days):

$$\begin{aligned} days &= MSF_d(1,2,3,5,8,15) \\ weeks &= MSF_w(5,9,18,31,56,75) \\ months &= MSF_m(22,47,99,147,214,313) \\ years &= MSF_y(224,581,1198,1948,3155,6509) \end{aligned}$$

We calculated the histogram of the given answers for each question, and we used the percentage of answers given to each option to create a combined membership function using the same parameters percentage extracted from each IV expression. For example, in question number 7 from Figure 3.11 the temporal context is “7 years" (or 2555 days). Supposing that received the following number of answers: 2 for “days", 3 for “weeks", 4 for “months", and 1 for “years" (a total of 10 answers), we used the corresponding IV definitions to combine a final membership function combining the parameters that define each IV expression in the same proportion of the percentage of given answers.

$$recently(2555) = 0.2 \times MSF_d + 0.3 \times MSF_w + 0.4 \times MSF_m + 0.1 \times MSF_y$$

which is equivalent to:

$$\begin{aligned} recently(2555) = MSF(&0.2 \times 1 + 0.3 \times 5 + 0.4 \times 22 + 0.1 \times 224, \\ &0.2 \times 2 + 0.3 \times 9 + 0.4 \times 47 + 0.1 \times 581, \\ &0.2 \times 3 + 0.3 \times 18 + 0.4 \times 99 + 0.1 \times 1198, \\ &0.2 \times 5 + 0.3 \times 31 + 0.4 \times 147 + 0.1 \times 1948, \\ &0.2 \times 8 + 0.3 \times 56 + 0.4 \times 214 + 0.1 \times 3155, \\ &0.2 \times 15 + 0.3 \times 75 + 0.4 \times 313 + 0.1 \times 6509) \end{aligned}$$

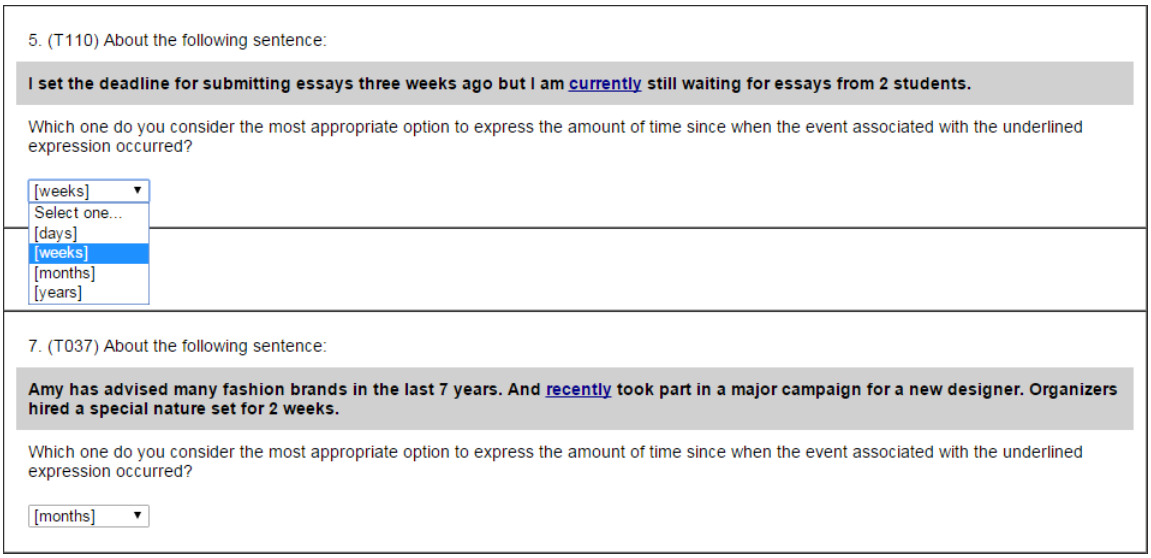


Figure 3.11: Example of questions covering PR imprecise timexes in English.

or:

$$recently(2555) = MSF(32, 80, 165, 264, 420, 802)$$

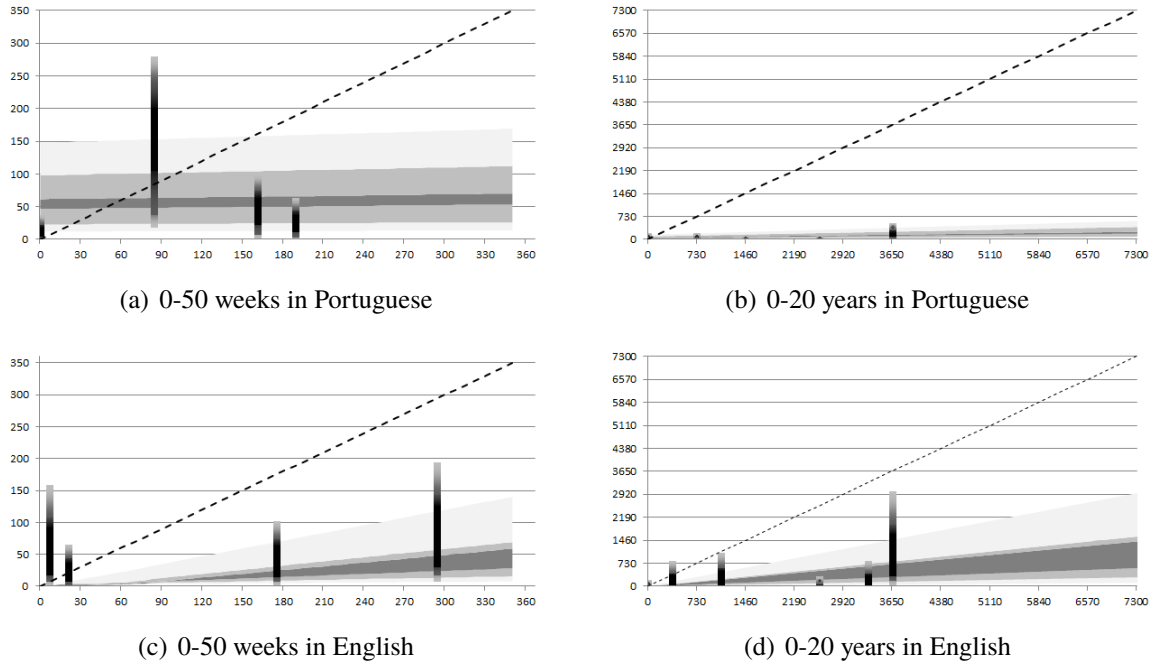
To calculate the linear regression model, we used the percentage of answers given for each PR question in order to produce a generic model based on the temporal context (in days). Figure 3.12 shows the models for two different periods (50 weeks and 20 years), including the resulted membership functions representation for each PR question in English and Portuguese.

Comparing PR imprecise expressions, we found $F1 = 0.391$ and $F1_{3D} = 0.304$ as the similarity between the linear regression models resulted for Portuguese and English.

3.6 Summary

In this chapter we described a series of evaluations that aims to verify the effectiveness of existing approaches to semantic analysis for temporal expression, event, and temporal relation extraction (SemEval - TempEval). We present two approaches we developed to time expression identification, as entered in to SemEval-2015 Task 6, Clinical TempEval. The first is a comprehensive rule-based approach (HINX) that favoured recall, and which achieved the best recall for time expression identification in Clinical TempEval. The second is an SVM-based system built using readily available components, which was able to achieve a competitive F1 in a short development time. We discussed how they perform relative to each other, and how characteristics of the corpus affect outcomes and the suitability of the two approaches.

Adapting annotation of temporal semantics to clinical notes is a significant and challenging task. Thus, we also detailed the results of a principled analysis of expert manual annotations of temporal expressions in the THYME schema over a corpus of clinical notes. Discrepancies between annotations and the guidelines were found in multiple categories. The spans or temporal expressions were not always correct. Ambiguity remained regarding the correct timex class, as happened also in TimeML. Wording in the guidelines was sometimes misinterpreted leading to non-markable timexes being annotated. Finally, as in TimeML, confusion appeared around the annotation of complex SET-type timexes and their quantifiers.



(a) 0-50 weeks in Portuguese

(b) 0-20 years in Portuguese

(c) 0-50 weeks in English

(d) 0-20 years in English

Figure 3.12: Hexagonal membership function model for PR imprecise timexes.

This data-driven analysis and its findings should help guide future temporal annotation efforts in the clinical domain.

As the main contribution in this thesis, we presented a methodology for the normalisation of imprecise timexes. It comprises different steps, from creating a set of questionnaires that are used as input and test data, pre-processing collected data, using linear regression and machine learning techniques in order to create imprecise representation models. We used F1-score to calculate and describe how similar two membership functions are, and proposed $F1_{3D}$ score as a complementary metric to describe whether the differences are more concentrated at the top or at the bottom. We also used F1-score and to choose the most suitable representation model for each kind if imprecise timex. We presented analysis for three kinds of imprecise timexes, but the approach could be used for all of the kinds of imprecise temporal expression that were presented. As a result of this methodology, imprecise timex can be represented in terms of numeric values, instead of described in terms of language structures as it happens when using current version of TimeML. Thus, normalised imprecise values can be used in logical and arithmetic operations, such as comparing imprecise timexes according to Allen's temporal relations, or using the imprecise normalised values as part of a search engine to improve the result of found extracted events.

Chapter 4

Similarity Search in the Information Extraction Process

The extraction of information from textual data sources is often performed using string similarity comparison algorithms to identify concepts from the free text [Godbole et al., 2010] when text is loaded with misspellings. String similarity metrics measure similarity between two text strings, and can be used to compare the elements from the input data source with an existing dictionary to identify a possible valid word for a misspelling.

The existing string similarity algorithms coupled with a supporting dictionary may be inefficient, in particular when the analysed text has spelling errors [Stvilia, 2007], because they may not necessarily handle specific aspects related to spelling errors, like phonetic errors. In these cases, it is necessary to use phonetic similarity metrics. Phonetics are language-dependent [Ladefoged and Maddieson, 1996] and solutions for this sort of problems must be designed for each specific language. In addition, similarity algorithms are often slow when executed over large databases, although fast search algorithms have been implemented.

Princeton WordNet (PWN) is a well-known lexical database, originally designed for the English language, that provides a combination of dictionary and thesaurus to support automatic text analysis coupled with artificial intelligence applications [Miller, 1995]. However, PWN had to be modified in order to support similarity search, as: (a) PWN does not include information about derivative words and the forms of irregular verbs; this problem is even greater when considering the variation of verb conjugation in different tenses (e.g. 67 variations for each verb in the Brazilian Portuguese language) [Ferreira, 2004]; and (b) PWN does not offer a repository structure to support string or phonetic similarity search.

In this chapter, we present an approach to fast phonetic similarity search (FPSS) over large repositories[1] [Tissot et al., 2014], coupled with a dictionary. Our solution has three main contributions. First, we present an indexed data structure called *PhoneticMap*, which is used by our novel fast similarity search algorithm. Second, we define a string similarity method that keeps the similarity higher for words with low differences. Specifically, it adds the notion of penalty, where the similarity value drops faster when the words have several differences. Finally, we integrate the previous contributions with Princeton WordNet (PWN) to implement the fast phonetic search.

We validate our approach through two sets of experiments. First, we compare our approach with existing methods and with our fast algorithm using phonetic maps. Second, we apply our solution in a use case for finding drug names and temporal tokens with misspelling

[1]This work has been published at DEXA'14: 25th International Conference on Database and Expert Systems Applications. September, 2014. Munich, Germany

errors in a set of more than 4 thousand medical records. The most important difference of our approach is the utilisation of phonetic information in an indexed structure. This structure is well adapted for calculating string and phonetic similarity between misspelt words.

In Section 4.1 we propose a string similarity function, a phonetic similarity function, and a method for searching for phonetic similarities over PWN-based repositories. Section 4.2 describes the experiment where the proposed methods were applied and shows the results obtained using Brazilian Portuguese words. Finally, in Section 4.3 presents the results from an experiment where we looked for misspelt variations of drug names and timexes in a set of medical records.

4.1 Fast Phonetic Similarity Search

Similarity search methods are essential to extract words from repositories with spelling errors. From a set of 157,064 words extracted from a set of 4,748 medical records written in Portuguese, we select a subset of 126,812 words with $length >= 5$ – avoiding abbreviations and acronyms – to conduct a simple experiment in which each word was submitted to an exact match over a dictionary. Only 40,212 words (31,71%) were found. Efficient inexact methods are important to be able to improve such results. However, when dealing with large repositories, it is also required to support a fast similarity search, i.e. for a given possible not well-written word, we want to find phonetically similar words. However, we want to avoid performing a full search in the repository.

4.1.1 String Similarity

We observed in initial experiments that traditional string distance metrics are not sufficient to reach the desired results when searching for words with orthographic errors. That led us to design a novel string similarity function to supply that shortcoming. We present a novel algorithm to calculate string similarity. The $String_{sim}$ function illustrated in Figure 4.1 measures similarity based on the percentage of characters of one word that can be found in the other one, also considering the position of matching characters and the difference in string sizes, obtaining a similarity value between 0 (completely different) and 1 (exactly equal).

The $String_{sim}$ function calculates the average between the percentage of w_1 characters found in w_2 and the percentage of w_2 characters found in w_1 (lines 1–6). *CharsFound* return the number of characters of the first parameter found in the second one, not taking into account the position in which the characters are found. For each character found, but not in the same string position, a reduction penalty is calculated based on the constant Ω (lines 3–5). *PositionPenalty* returns the number of characters of the first parameter found in the second one but not in the same string position. The penalty is calculated based on Ω and guarantees, for example, that strings “ba" and “baba" will NOT result in a $similarity = 100\%$. When the lengths of both strings (S_{MAX} and s_{min}) are different, there is an adjustment in order to provide another penalty in the similarity level, based in the difference on the length of words and the factor Υ (lines 7–15). Ω (=0.975) and Υ (=0.005) were empirically defined after testing the proposed function in an application that searches for similar names of people and companies. Ω and Υ were manually adjusted based on a list of known pairs of names that should or should not be considered similar in the result of each search, looking for a better precision/recall based on that list. Those tests were not exhaustive, and they surely need more investigation, particularly how different sets of parameter can better fit into different domains and languages.

in:	w_1 String, w_2 String
out:	$similarity$ Number
1:	$g_1 \leftarrow CharsFound(w_1, w_2)$;
2:	$g_2 \leftarrow CharsFound(w_2, w_1)$;
3:	$\Omega \leftarrow 0.975$;
4:	$p_1 \leftarrow \Omega^{PositionPenalty(w_1,w_2)}$;
5:	$p_2 \leftarrow \Omega^{PositionPenalty(w_2,w_1)}$;
6:	$similarity \leftarrow avg(g_1 \times p_1, g_2 \times p_2)$;
7:	$\Upsilon \leftarrow 0.005$;
8:	$S_{MAX} \leftarrow MAX(length(w_1), length(w_2))$;
9:	$s_{min} \leftarrow min(length(w_1), length(w_2))$;
10:	if $(S_{MAX} > s_{min})$ then
11:	$b \leftarrow 1 + (S_{MAX} - s_{min}) \times \Upsilon$;
12:	$f \leftarrow ln(S_{MAX} - s_{min} + 1)$;
13:	$c \leftarrow \frac{S_{MAX} - s_{min}}{2}$;
14:	$similarity \leftarrow similarity \times (\frac{1}{(b^f)^c})$;
15:	end if;
16:	return $similarity$;

Figure 4.1: $String_{sim}$: A proposed string similarity function pseudocode.

Figure 4.2 shows the differences in the result of $String_{sim}$ function compared to two other similarity functions for 36 randomly selected pairs of similar words in Portuguese – results are ordered according to the descending order of Edit Distance function (Levenshtein). Depending on the words, each distance metric can result in a relatively greater or lesser similarity value, e.g. for some pairs of words considered less (or more) similar using Jaro-Winkler function, $String_{sim}$ function resulted a greater (or lesser) similarity value.

4.1.2 Phonetic Similarity

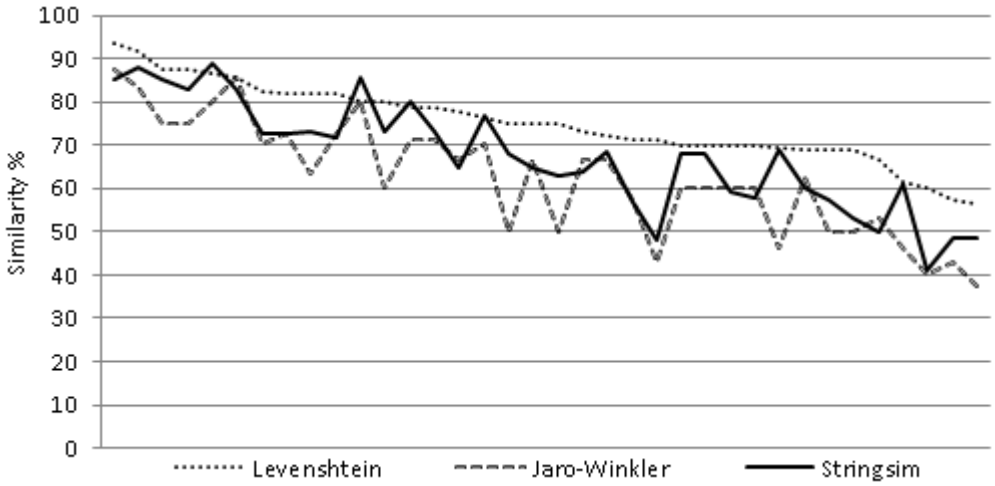
When considering phonemes, a straightforward string comparison of characters may not be enough. In order to support indexing phonemes for a fast search, we present a structure called *PhoneticMap* and we define the *PhoneticMap Similarity*.

Definition 4.1 (PhoneticMap) *Given a word (string) w consisting of a sequence of letters (as symbols and digits usually fall outside this scope), the generic function $PhoneticMap(w)$ is a function that results in a PhoneticMap tuple $M = (w, P, D)$, where: w is the word itself, $P = \{p_1, p_2, \ldots, p_n\}$ is a set of n phonetic variations of word w, and $D = \{d_1, d_2, \ldots, d_n\}$ is a set of n definitions, where d_i is the description of variation p_i.*

Definition 4.2 (PhoneticMap Similarity) *$PhoneticMapSim(M_1, M_2)$ is a generic function that returns a similarity value (ranging from 0=different to 1=equal) between PhoneticMaps M_1 and M_2.*

As $PhoneticMap(w)$ and $PhoneticMapSim(M_1, M_2)$ are language-dependent, it is even possible to create more than one instance to each function for different languages [2].

[2]The Brazilian Portuguese consonant phonemes can be classified according to the manner of articulation and the place of articulation of vocal chords[Catarino, 1999]. The manner of articulation codes comprise: A (nasal) for {/m/,



* x-axis represents each compared pair of words

Figure 4.2: Comparing results between similarity functions.

Table 4.1: $PhoneticMap_{PT}$("arrematação").

Entry i	Definition d_i	Phonetic variation p_i
w	Word	*arrematação*
1	Word with no accents	*arrematacao*
2	Word phonemes	*aRematasao*
3	Vowel phonemes only	*aeaaao*
4	Vowel phonemes (reverse)	*oaaaea*
5	Consonant phonemes	*Rmts*
6	Consonant phonemes (reverse)	*stmR*
7	Articulation manner	*EABC*
8	Articulation manner (reverse)	*CBAE*
9	Articulation point	*FACD*
10	Articulation point (reverse)	*DCAF*

We develop two variations corresponding to the *PhoneticMap* and *PhoneticMap Similarity* functions to support the Brazilian Portuguese language: $PhoneticMap_{PT}(w)$ and $PhoneticMapSim_{PT}(M_1, M_2)$.

The function $PhoneticMap_{PT}(w)$ returns a map of 11 entries. Table 4.1 describes each entry and shows an example generated for a Brazilian Portuguese word. This structure can be adapted for different languages.

The function $PhoneticMapSim_{PT}(M_1, M_2)$, as defined in Equation 4.1, calculates the phonetic similarity between PhoneticMaps M_1 and M_2 as the string similarity weighted average between some phonetic variations of M_1 and M_2. The same list of known pairs used to adjust

/n/, /nh/}; B (stop) for {/b/, /k/, /d/, /g (gue)/, /p/, /t/}; C (fricative) for {/s/, /f/, /j/, /v/, /x/, /z/}; D (constricting lateral) for {/l/, /lh/}; and E (constricting vibrant) for {/r/, /R/}. The place of articulation codes comprise: A (bilabial) for {/m/, /p/, /b/}; B (labio-dental) for {/f/, /v/}; C (linguo-dental) for {/t/, /d/}; D (alveolar) for {/l/, /lh/, /z/, /s/, /n/, /nh/, /r/}; E (palatal) for {/j/, /x/}; and F (velar) for {/k/, /R/, /g (gue)/}. Such codes were used to generate phonetic variations 7–10 based on variation 5 (Table 4.1).

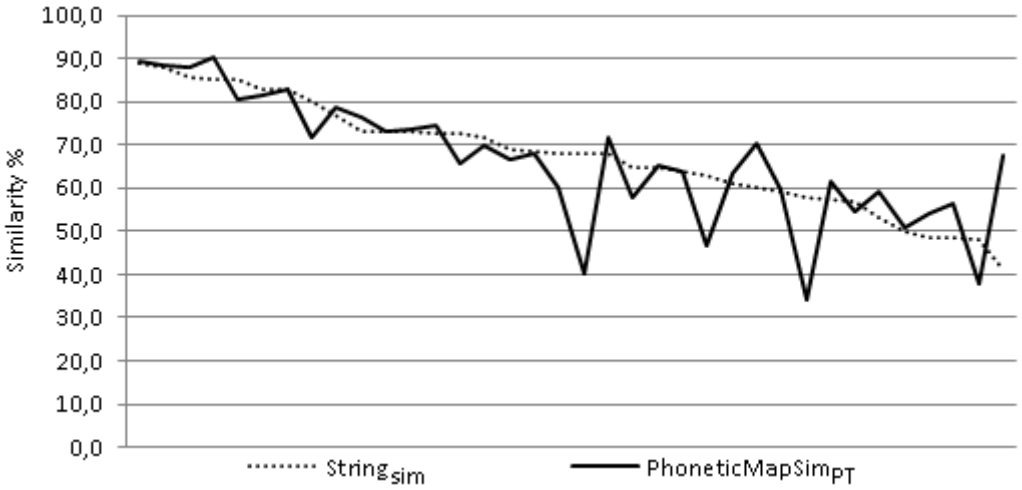
StringSim factors were used to empirically adjust weights used in $PhoneticMapSim_{PT}$. We tested different sets of values between 0 and 10 in order to improve precision/recall. The final set of weights gives more importance to similarities of consonant phonemes.

$$PhoneticMapSim_{PT}(M_1, M_2) = \frac{1 \times S_w + 2 \times S_{(1)} + 5 \times S_{(2)} + 1 \times S_{(3)} + 3 \times S_{(5)} + 2 \times S_{(7)} + 2 \times S_{(9)}}{1 + 2 + 5 + 1 + 3 + 2 + 2} \quad (4.1)$$

where:

$S_w = String_{sim}((M_1.w, M_2.w)$
$S_{(i)} = String_{sim}((M_1.p_i, M_2.p_i)$

Figure 4.3 shows the string and phonetic similarities obtained by the $String_{sim}$ and the $PhoneticMapSim_{PT}$ functions, based on a list of 36 pairs of similar words. We can observe that, depending on the words, the phonetic similarity can be greater or lesser than the string similarity – results are ordered according to the descendant order of $String_{sim}$ function.



* x-axis represents each compared pair of words

Figure 4.3: Comparing results between string and phonetic similarity functions.

4.1.3 Phonetic Search

To perform a fast phonetic similarity search (FPSS), we propose a method for indexing PhoneticMaps using single column indexes in a relational database. FPSS must locate phonetically similar words in the repositories based on the indexed phoneme variations, returning not only similar words but also the similarity level of each one.

Definition 4.3 (Fast Phonetic Similarity Search) *Given a word w and a minimum desirable similarity level l, $PhoneticSearch(w, l)$ is a generic function that results a set of tuples (r, s), where r is a phonetically similar word, and s is the similarity level obtained between $PhoneticMap(w)$ and $PhoneticMap(r)$, where $s \geq l$. Similarity level ranges from 0 to 1.*

Each *Phonetic Variation* (01–10) is indexed to support a fast search over each column. FPSS is performed using the pseudocode described in Figure 4.4.

in:	$word$ String, $minSimLevel$ Number, $prefix$ Integer default 0, $suffix$ Integer default 0
out:	$resultSet$ Dataset
1 :	$pm \leftarrow PhoneticMap_{PT}(word)$;
2 :	if $minSimLevel = 1$ then
3 :	$resultSet \leftarrow DBPhoneticMapSearch(0, pm.w)$;
4 :	else;
5 :	$resultSet \leftarrow$ $DBPhoneticMapSearch(1, pm.p_1) \cup$ $DBPhoneticMapSearch(2, pm.p_2) \cup$ $DBPhoneticMapSearch(3, pm.p_3, suffix) \cup$ $DBPhoneticMapSearch(5, pm.p_5, suffix) \cup$ $DBPhoneticMapSearch(7, pm.p_7, suffix) \cup$ $DBPhoneticMapSearch(9, pm.p_9, suffix)$;
6 :	if $prefix > 0$ then
7 :	$resultSet \leftarrow resultSet \cup$ $DBPhoneticMapSearch(4, pm.p_4) \cup$ $DBPhoneticMapSearch(6, pm.p_6, prefix) \cup$ $DBPhoneticMapSearch(8, pm.p_8, prefix) \cup$ $DBPhoneticMapSearch(10, pm.p_{10}, prefix)$;
8 :	end if;
9 :	foreach ($findWord$ in $resultSet$)
10 :	if $PhoneticMapSim_{PT}(pm, PhoneticMap_{PT}(findWord))$ $< minSimLevel$ then
11 :	$resultSet.remove(findWord)$;
12 :	end if;
13 :	end if;
14 :	return $resultSet$;

Figure 4.4: $PhoneticSearch_{PT}$ pseudocode.

$PhoneticSearch_{PT}$ returns a set of similar words in Portuguese for a given input $word$, considering the minimum desirable similarity level $minSimLevel$. Additional parameters $prefix$ and $suffix$ set the number of extended consonant phonemes that can be considered as prefix and suffix when searching for similar words. $prefix$ and $suffix$ have default values 0 (zero). When $prefix > 0$, then $PhoneticSearch_{PT}$ uses the reverse indexed PhoneticMaps entries to locate similar words (entries 4, 6, 8 and 10 described in Table 4.1). Function $DBPhoneticMapSearch(i, v, e)$ finds records in the *PhoneticMap* table, searching for $PhoneticMap$ entry i equals to value v (exact match), or entry i like value v with up to e characters added ("like" match), when $e > 0$. $PhoneticSearch_{PT}$ results a exact match when $l = 1$ (lines 2–3). Otherwise, it creates a dataset combining results of different $DBPhoneticMapSearch$ executions (line 5). In lines 6–8, phonetic variations 4, 6, 8, and 10 are used whether it is necessary to perform search over the reverse PhoneticMap entries ($prefix > 0$). After creating a result set of candidate words, the phonetic similarity between each found word and the search word is calculated (line 10). Words that does not satisfy the minimum similarity level $minSimLevel$ are removed from the result set (lines 10-11).

To support FPSS, we extended the PWN repository with the *PhoneticMap* table that stores PhoneticMap entries for a specific language, as illustrated in the PWN Schema subset in

Figure 4.5. Thus, it was possible to combine the FPSS with the PWN Repository to retrieve additional information about the search words, as part of speach (POS) tags.

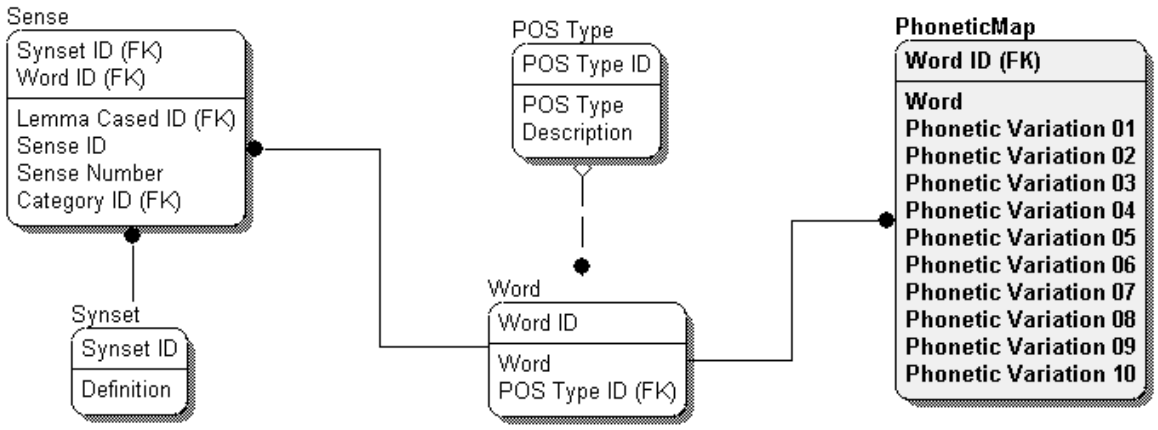


Figure 4.5: Extended Wordnet Subset.

Even though it is presented as an instance for Brazilian Portuguese language, this approach is tailored to adapt phonetic matching to use over large repositories and for different languages, such as English and Spanish, since one can define a new $PhoneticMap$ structure for an specific language, and instantiate the $PhoneticMapSim$ and $PhoneticSearch$ functions for such language.

4.2 Experimental comparison

In this section we describe the experiments conducted to validate our approach. We perform different comparison studies to verify the effectiveness of the solution. First, we compare our String Similarity algorithms with two well-known ones. Secondly, we compare the performance of a full search method with a search using the indexed PhoneticMaps. Finally, we describe precision and recall results of our fast phonetic search solution.

4.2.1 String Similarity

Different algorithms have different behaviour when decreasing the similarity as the words become more different from each other. We compared $String_{sim}$ against normalised versions of Edit Distance (by Levenshtein [Levenshtein, 1966]) and Jaro-Winkler distance [Cohen et al., 2003]. In Figure 4.6 we illustrate how each function decreases the similarity as the words become more different from each other (some similarity results are detailed in Table 4.2). In this case we used only a combination of letters, for didactic purposes. $String_{sim}$ keeps the similarity higher as there are more characters in common between strings with small differences in size. Comparing strings “ab" and “abab", Edit Distance results in a similarity of 50% while $String_{sim}$ is 96.9%. As the difference in length of strings becomes larger, $String_{sim}$ tends to reduce the similarity more sharply, getting close to 0% faster. $String_{sim}$ also has the ability to distinguish different characters when comparing strings. For example, Edit Distance yield the same similarity (50%) when comparing (“ab" × “abab") or (“ab" × “abcd"). The same is not true when using $String_{sim}$: $String_{sim}$(“ab”, “$abab$”) = 96.9% and $String_{sim}$(“ab”, “$abcd$”) = 74.2%.

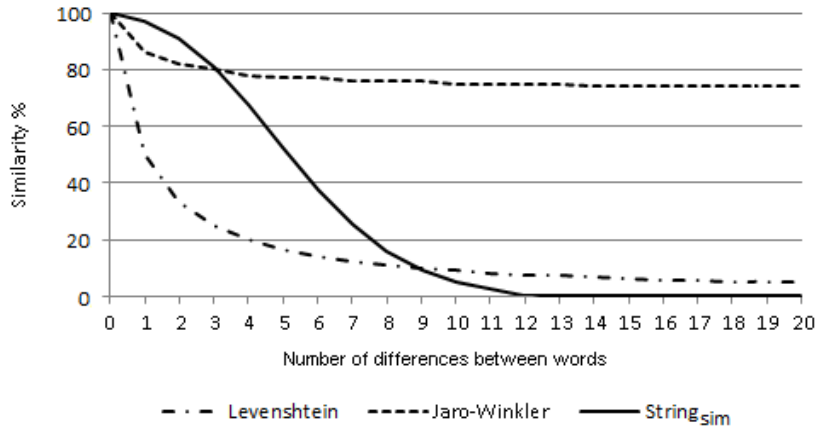


Figure 4.6: Similarity functions behaviour.

Table 4.2: Comparing similarity functions.

#	$Word_1$	$Word_2$	Levenshtein	Jaro-Winkler	$String_{sim}$
0	ab	ab	100.0%	100.0%	100.0%
1	ab	abab	50.0%	86,0%	96.9%
2	ab	ababab	33.3%	82.0%	91.0%
...	...	...	...	...	...
10	ab	ab ... ab (10x)	10.0%	76.0%	9.1%
...	...	...	...	...	...
20	ab	ab ... ab (20x)	5.0%	74.0%	0.0%

= number of repetitions of “ab" in $Word_2$

We also performed an experiment to verify the efficiency of $String_{sim}$ in automatic error correction compared with other functions, comprising the following steps:

- We extracted a set of 3,933 words (non-clinical domain) containing spelling errors from a sample of medical records written in Portuguese. Each word was manually annotated with the correct spelling form (reference word).
- We used the $String_{sim}$ function to search for the 10 most similar words for each incorrect word, based on the returned similarity values. We performed searches using a Brazilian Portuguese version of PWN dictionary (containing 798,750 distinct words, verb conjugation derivatives). The result sets for each word were ranked from 1 (most similar) to 10 (less similar).
- We saved the rank in which each reference word was found in each result set — $Rank = NotFound$ when the reference word is not in the resulted record set.
- The two previous steps were repeated using Edit distance and Jaro-Winkler functions — thus, for each misspelt word, we store the rank of the reference word for each one of the three similarity functions.

- Lastly, we compared the results of $String_{sim}$ against Edit distance and Jaro-Winkler, as shown in Tables 4.3 and 4.4. These tables show that $String_{sim}$ had more reference words with top-1 ranking, which is the objective of the approach.

Table 4.3 compares $String_{sim}$ and Edit Distance. In 75.5% of cases (2,970 words), both functions find the reference word in the dictionary as a top-1 ranking (the most similar). For the remaining cases, where the reference words were found amongst the ranked top-10 search results, $String_{sim}$ performs better (finds the reference word in a better rank) than Edit Distance in 16.9% of cases (666 searches with better ranking) while Edit Distance is better than $String_{sim}$ in only 5.8% (230 searches).

Table 4.3: $String_{sim}$ (SS) x Edit Distance (ED).

SS	ED Rank					
Rank	1	2	3	4-5	6-10	*Not Found*
1	2970	420	51	30	25	26
2	127	51	37	15	18	13
3	32	12	8	8	3	7
4-5	17	8	7	4	6	3
6-10	14	1	2	4	4	0
Not Found	2	0	0	1	1	6

Table 4.4 compares $String_{sim}$ and Jaro-Winkler distance. In 68.0% of cases (2,675 words), both functions find the reference word as a top-1 ranking. For other cases, $String_{sim}$ performs better than Jaro-Winkler in 23.7% of cases (934 searches) while Jaro-Winkler is better than $String_{sim}$ in only 4.9% (193 searches).

Table 4.4: $String_{sim}$ (SS) x Jaro-Winkler (JW).

SS	JW Rank					
Rank	1	2	3	4-5	6-10	*Not Found*
1	2675	360	119	107	92	169
2	102	49	25	19	22	44
3	20	19	11	8	4	8
4-5	14	8	4	8	4	7
6-10	4	9	3	1	3	5
Not Found	1	1	1	1	2	4

4.2.2 Full and Fast Similarity Search

We compared the performance of full and fast similarity search methods, using the full set of words extracted from a well-known Brazilian Portuguese dictionary [Ferreira, 2004]. The steps to perform the experiment are described as follows[3]:

[3]We implemented our solution using an instance of Oracle database version 11g running over a Intel(R) Core(TM) i5 2.50 GHz with 8GB RAM.

- A WordNet repository was created in a relational database and it was populated with a total amount of 798,750 distinct words and verb conjugation derivatives;
- one PhoneticMap for each dictionary entry was created with the function $PhoneticMap_{PT}$, populating the *PhoneticMap* table. The table was indexed with 11 single-column indexes – one for the *Word* column and one for each *Phonetic Variation*;
- The same 3,933 words containing spelling errors used in the previous experiment were applied to the search methods;
- A *Full Search* was executed – each input word was compared with each dictionary entry using the $String_{sim}$ (Figure 4.1), searching for words with a similarity level ≥ 0.8; the search time spent and the number of found words were computed in the result – the $PhoneticMapSim_{PT}$ function was not used in the *Full Search* due to its processing time (over 60 seconds to perform each search).
- A *Fast Search* was executed – each input word was submitted twice to $PhoneticSearch_{PT}$, with two different set of parameters: a) similarity level ≥ 0.9, and parameters p and s both equal to 0 (similar words might have the same number of consonant phonemes); and b) similarity level ≥ 0.8, and parameters p and s both equal to 1 (similar words could have one additional consonant phonemes as prefix or suffix);
- *Full Search* and *Fast Search* results were compared based on (a) the total amount of spent time to execute each search, and (b) the number of words obtained as the search result.

4.2.3 Comparing Results

Due to the indexed PhoneticMap structure, we observed that *Fast Search* can be 10-30 times faster than *Full Search* (Table 4.5). However, it must be clear that *Fast Search* does not return the same set of similar words, comparing to the result of *Full Search*. Although a *Full Search* is complete in terms of the resulting words, both search methods did not use the same similarity function — *Full Search* was performed with $String_{sim}$, and *Fast Search* used the phonetic similarity metric $PhoneticMapSim_{PT}$, i.e. using the *PhoneticMap*s.

Table 4.5: Average spent time (in seconds) to execute word search.

Method	Similarity Level	Spent Time (average)	Words Found (average)
Full	≥ 0.80	4.92 seconds	29.95
Fast	≥ 0.80	0.49 seconds	62.31
Fast	≥ 0.90	0.17 seconds	6.28

Even though the fast search method returns a different set of words, the $PhoneticSearch_{PT}$ function is able to find the reference word for each spelling error. Table 4.6 compares the accuracy of $PhoneticSearch_{PT}$ (PS) against $String_{sim}$ (SS). In 80.6% of cases (3.170 words), both functions find the reference word as a top-1 ranking. $String_{sim}$ performs better in 10.2% (402 searches) while PS is better in 8.1% (317 searches) of the remaining cases where the reference words were found amongst the ranked top-10 search results. Tables 4.6 shows the amount of cases in which the $PhoneticSearch_{PT}$ results were better than $String_{sim}$.

The phonetic approach compensates that loss – around 2% of the cases in terms of top-1 ranking – with a better time response (as shown in Table 4.5) when searching over large repositories.

Table 4.6: $PhoneticSearch_{PT}$ x $String_{sim}$.

PS Rank	SS Rank 1	2	3	4-5	6-10	*Not Found*
1	3170	189	50	31	14	5
2	143	37	10	7	1	0
3	57	13	1	2	0	1
4-5	46	9	6	4	5	0
6-10	47	6	0	0	2	1
Not Found	59	7	3	1	3	3

We analysed our approach through precision, recall, and F1-score relevance measures [Davis and Goadrich, 2006]. In our experiment, the concepts TP, FP and FN can be defined according to the results obtained by the *Fast Search* with respect to the results previously performed by the *Full Search*. It means that, for each word w submitted to the $PhoneticSearch_{PT}$, we define $TP(w)$, $FP(w)$ and $FN(w)$ (Equations 4.2, 4.3 and 4.4), where $FastSearch(W, \alpha)$ represents the list of words returned by a *Fast Search* method performed to a given word w using a similarity level α, and $FullSearch(W, \beta)$ is equivalent to the list of words returned by a *Full Search* method performed to a given word w using a similarity level β.

$$TP(w) = count\big(FastSearch(w, \alpha) \cap FullSearch(w, \beta)\big) \tag{4.2}$$

$$FP(w) = count\big(FastSearch(w, \alpha) - FullSearch(w, \beta)\big) \tag{4.3}$$

$$FN(w) = count\big(FullSearch(w, \beta) - FastSearch(w, \alpha)\big) \tag{4.4}$$

Table 4.7 shows the average precision and recall, calculated based on the precision and recall of each word submitted to the *Fast Search*, compared to the *Full Search* with $\beta = 0.9$. Lower values are observed in column *Avg Precision* for $\alpha < 0.95$ (value of FP is high). It means that $FastSearch$ returns more similar words then $FullSearch$, even considering $\alpha > \beta$. Column *Avg Recall* highlights the fact of FN is, in average, around 25% of TP, what means that 25% of words returned by $FullSearch$ do not appear in the $FastSearch$ results. This loss also draws that not all words returned as "similar" by the $String_{sim}$ function in the *Full Search* method are considered phonetically similar by the $PhoneticMapSim_{PT}$ function in the *Fast Search* method. The results obtained with full (string similarity) and fast (phonetic similarity) methods are complementary. A hybrid approach can generate better results than using one single alternative.

To illustrate practical differences between the full and fast methods, Table 4.8 shows the result of full and fast searches with spelling errors. The word "*confidencial*" (*confidential*) was changed to "***b****onfidenci****e****l*" to be applied in the search methods handling two spelling errors. While Full Search took 33.9 seconds to scan almost 800,000 dictionaries entries and to produce the result, Fast Search spent 0.4 seconds to achieve it. In Fast Search, original word ("*confidencial*") is returned as top-1 result against 4^{th} ranked position in Full Search, even string similarity (86%) is subtly greater than phonetic similarity (85,9%).

Table 4.7: Precision, Recall and F1-score results ($\beta = 0.9$).

α	**Avg Precision**	**Avg Recall**	**F1-score**
0.80	0.1838%	0.8616%	0.3029
0.85	0.3014%	0.8529%	0.4454
0.90	0.4836%	0.7923%	0.6006
0.95	0.7342%	0.7028%	0.7182

Table 4.8: Examples of Full and Fast Search results for "***b**onfidenci**e**l*"*.

Full String Similarity Search		*Fast Phonetic Similarity Search*	
Found Word	Similarity (%)	Found Word	Similarity (%)
confidencie	89.0	**confidencial**	**85.9**
confidenciei	88.0	confidencie	85.5
confidencio	86.0	confidenciei	84.8
confidencial	**86.0**	confidencies	84.7
confidencies	85.0	confidencieis	84.1
confidenciem	85.0	confidenciem	83.0

* "***b**onfidenci**e**l*" = "*confidencial*" with two spelling errors.

In Portuguese, words with completely different meanings can be quite similar in writing, like "*velocidade*" (speed), "*voracidade*" (greed), and "*veracidade*" (truthfulness). Table 4.9 shows the search result of full and fast methods for a misspelt word ("*verocidade*"), which became similar to the first three mentioned. Full Search spent 36.6 seconds to return the result, while Fast Search spent only 0.3 seconds. We can also highlight the differences in rank positions of each word when comparing both search results.

4.3 Misspelt Drug Names and Timexes

In this section we describe the case study developed to show how String and Phonetic similarity metrics can be combined in a hybrid solution to identify names of drugs within 4,748 medical records written in Portuguese. We have a base list of 5,535 drug names extracted from the *InfoSaude* system. In this experiment we try to find the best combination of boundaries for using both metrics on identifying misspelt names of drugs.

According to [Senger et al., 2010], spelling errors of generic drug names can occur in up to one out of six entries in electronic drug information systems. Such errors are likely to be responsible for up to 12% of adverse drug events, mainly caused by errors during transcription of prescriptions, illegible prescriptions, or drug name confusion. Due to such errors' frequency and the relevance of drug information in clinical tasks, spelling error-tolerant engine systems and automatic spelling correction can be useful to health care professionals.

Since July 2013 the Brazilian government has been trying to address the shortage of doctors, especially in the inner cities and the outskirts of large cities in Brazil, through the hiring

Table 4.9: Sample of Full and Fast Search results for "*verocidade*".

Full String Similarity Search		Fast Phonetic Similarity Search	
Found Word	Similarity (%)	Found Word	Similarity (%)
veracidade	**90.0**	**veracidade**	**94.1**
veridicidade	89.0	versidade	93.3
ferocidade	88.0	**voracidade**	**92.0**
velocidade	**87.0**	ferocidade	90.1
verticidade	86.0	ferocidades	88.8
voracidade	**85.0**	**velocidade**	**86.6**
atrocidade	85.0	velocidades	85.5
heroicidade	84.0	feracidade	84.6
ferocidades	84.0	serosidade	83.4
verdadeiro	83.0	verbosidade	83.1

of doctors from other countries[4]. With the addition of doctors from other countries working in the health system (especially from countries in South and Central America where people naturally speak Spanish), a larger number of spelling errors have been found in medical records stored in the *InfoSaude* system. Such errors occur mainly due to the similarity of the Portuguese language with other Latin languages, such as Spanish and Italian.

Furthermore, the textual content of the medical records does not go through any kind of review. Thus, it is common to find a number of spelling and phonetic errors that could harm any further analysis of it. An extraction system process must deal with this problem to avoid information loss. The proposed phonetic similarity search method can be applied to identify possible names of drugs when they are incorrectly written. However, as spelling errors are recognised based on a similarity value, it is necessary to set an appropriate threshold to determine whether or not a misspelt word corresponds to a drug name.

We divided the experiment into three steps: a) we identified the most cited drugs on the input corpus that were correctly spelt – these drugs were used as basis for the similarity search, so it was not necessary to look over the full input corpus; b) we implemented an algorithm to find the appropriate similarity thresholds to identify misspelt drug names; and c) we execute our similarity search method over the medical record set. We explain these steps below.

First, we performed an exact match search, finding 516 drug names. From this result we extracted the 20 most cited drugs in the text, which are listed in Table 4.10.

Secondly, we used drugs listed in Table 4.10 to establish the appropriate string and phonetic similarity thresholds. Such thresholds are applied in the information extraction process to determine whether a candidate similar word corresponds to a drug name. Inappropriate low threshold values may return too many results, including words with low similarity values that do not correspond to a drug name. In contrast, high threshold values may exclude possible valid misspelt drug names from the final matching. The method used to find the most suitable string and phonetic similarity thresholds are described below:

- We selected a list of candidate words (similar words) for each drug, extracting from text all words that had at least 3 consonantal phonemes matching the phonemes in each drug name

[4]Brazilian Ministry of Health: Programa Mais Medicos (More Doctors Program). http://portalsaude.saude.gov.br/index.php/cidadao/acoes-e-programas/mais-medicos (Accessed: Jun, 2015)

Table 4.10: Most cited drug names in the input medical records.

Drug	Occurrences	Drug	Occurrences
fluoxetina	18624	clorpromazina	4226
paracetamol	8697	enalapril	4144
diazepam	8474	imipramina	4135
amitriptilina	8463	sinvastatina	3862
omeprazol	7825	carbamazepina	3853
dipirona	7320	amoxicilina	3716
glicose	5721	ibuprofeno	3714
captopril	5383	metformina	3467
insulina	5290	risperidona	3464
nimesulida	4228	atenolol	3224

– for smaller names of drugs we considered the formula “at least (number of consonantal phonemes - 2)". The returned list of similar words was manually analysed to apply a filter that consider only words with $String_{sim}$ greater than 0.6. The final result is a list of 1,791 distinct candidate words for 20 drug names – an average of 90 similar candidate words per drug.

- We analysed the list of candidate words to identify which corresponded or not to a drug name. As a result, we manually annotated 938 positive matches and 853 negative matches. We also annotated each positive and negative match with the presented string and phonetic similarity measures.

- We used the annotated list to perform a search for the best similarity thresholds. The list of 20 drugs was split into 2 groups of 10 drugs to be used as a training and a validation sets, respectively. We combined all string and phonetic similarity values from the list (we also considered phonetic similarity as the only threshold parameter) to perform an exhaustive search to find the most appropriate similarity threshold values. The pseudocode is presented in Figure 4.7.

The pseudocode is an exhaustive search for the best pair of phonetic and string similarity thresholds. The input comprises two manually annotated lists ($trainSet$ and $validSet$) – containing names of drugs and candidate similar words with the corresponding positive or negative match flag – and a list with 7,730 pairs of possible string and phonetic threshold values. 660 pairs of similarity values contain $stringSim = 0$, i.e. a possible solution considering phonetic similarity as the only threshold. Finally, for each possible threshold values, the algorithm calculates $Precision$, $Recall$, and $F1$ for each set of 10 drugs ($trainSet$ – lines 2-7 – and validSet – lines 8-13). The final thresholds are updated each time both $F1_{train}$ and $F1_{valid}$ simultaneously achieve better values – lines 14-19.

After executing the described pseudocode on the data extracted from the medical record set, we concluded that a hybrid solution considering both phonetic and similarity thresholds is better to identify names of drugs. The hybrid solution means that we use a smaller phonetic threshold to perform a fast similarity search that result more similar words, complemented with a filter that uses a string similarity threshold. Table 4.11 describes the final results.

Lastly, we used the defined thresholds as a filter to find which pairs of candidate word and drug name were valid. However, even after using a filter to select the desirable similarity

```
in:   List trainSet, validSet, possibleThresholdSet
out:  Number phoneticSimilarityThreshold ← 0, stringSimilarityThreshold ← 0
var:  Number bestF1_train ← 0, bestF1_valid ← 0;
1:    for (each t in possibleThresholdSet) loop
        // Precision,Recall,F1 for trainSet
2:      TP_train ← CalcTruePositives(trainSet, t.phoneticSim, t.stringSim);
3:      FP_train ← CalcFalsePositives(trainSet, t.phoneticSim, t.stringSim);
4:      FN_train ← CalcFalseNegatives(trainSet, t.phoneticSim, t.stringSim);
5:      Precision_train ← TP_train/(TP_train + FP_train);
6:      Recall_train ← TP_train/(TP_train + FN_train);
7:      F1_train ← 2 * (Precision_train * Recall_train)/(Precision_train + Recall_train);
        // Precision,Recall,F1 for validSet
8:      TP_valid ← CalcTruePositives(validSet, t.phoneticSim, t.stringSim);
9:      FP_valid ← CalcFalsePositives(validSet, t.phoneticSim, t.stringSim);
10:     FN_valid ← CalcFalseNegatives(validSet, t.phoneticSim, t.stringSim);
11:     Precision_valid ← TP_valid/(TP_valid + FP_valid);
12:     Recall_valid ← TP_valid/(TP_valid + FN_valid);
13:     F1_valid ← 2 * (Precision_valid * Recall_valid)/(Precision_valid + Recall_valid);
        // Update Thresholds
14:     if (F1_train > bestF1_train) and (F1_valid > bestF1_valid) then
15:       bestF1_train ← F1_train;
16:       bestF1_valid ← F1_valid;
17:       phoneticSimilarityThreshold ← t.phoneticSim;
18:       stringSimilarityThreshold ← t.stringSim;
19:     endif;
20:   end loop;
21:   return (phoneticSimilarityThreshold, stringSimilarityThreshold);
```

Figure 4.7: A pseudocode to find similarity thresholds.

values, we found candidate words that were similar to more than one drug at the same time. This happened due to the fact that different drugs also have similar names. To solve this issue, we used the phonetic similarity as a final filter – for each candidate word similar to more than one drug name, we considered the pair with the higher phonetic similarity value as the correct match.

As a final result of this experiment, it was possible to identify a total of 1.442 misspelt words corresponding to 409 different drug names. Table 4.12 shows the drug names (but the ones used in the training and validating set) in which the greatest number of misspelt forms were found.

A hybrid solution was able to deal with both phonetic and spelling errors. When combining both String and Phonetic Similarity functions we found a set of thresholds that reached better precision and recall when looking for misspelt drug names. In the hybrid approach, the threshold for the Phonetic Similarity is lesser than that one found testing an approach that uses only the Phonetic Similarity function. Such difference is compensated by the String Similarity threshold, used as a complementary filter.

A similar experiment was performed using a set of 48 temporal tokens in Portuguese. Temporal tokens comprise words that can be used to identify temporal concepts, such as temporal granularities, periods of the day, names of months, days of week, names of seasons, and words that represent past, present and future references. Table 4.13 depicts the number of misspelt variations found for each temporal token.

Table 4.11: Best combination of string and phonetic thresholds.

	Parameter	Value
Training Set	Number of true positives	417
	Number of false positives	31
	Number of false negatives	25
	Precision	0.93080
	Recall	0.94344
	F1	0.93708
Validation Set	Number of true positives	477
	Number of false positives	39
	Number of false negatives	19
	Precision	0.92442
	Recall	0.96169
	F1	0.94269
Thresholds	Phonetic similarity	0.844
	String similarity	0.831

However, when trying to apply the phonetic and string similarity thresholds described in Table 4.11 (phonetic similarity threshold = 0.844 and string similarity threshold = 0.831) to select the possible misspelt temporal tokens, we observed an unexpected number of false positives/negatives.

Although the threshold parameters were efficient for selecting misspelt drug names, they were not adequate for selecting misspelt temporal tokens, possibly due to the length (string size) of temporal tokens comparing to the length drug names. Whilst drug names are in average 10.5 characters long, temporal token are in average no longer than 6.4 characters. That can indicate threshold parameters should be adapted for different string size ranges. That led us to manually select the the pair of misspelt and correct words for each temporal token, discharging those incorrect pairs, adapting both threshold values to fit the temporal token set (phonetic similarity threshold = 0.789 and string similarity threshold = 0.835).

4.4 Summary

In this chapter, we propose a string similarity function, a phonetic similarity function, and a method for searching for phonetic similarities over PWN-based repositories. Our approach of fast phonetic similarity search (FPSS) over large repositories has three main contributions: a) an indexed data structure (*PhoneticMap*), b) a novel string similarity algorithm (*StringSim*), and c) we integrated the previous contributions with Princeton WordNet (PWN) to implement the fast phonetic similarity search. The most important difference of our approach is the utilisation of phonetic information in an indexed structure.

We validated our approach through two sets of experiments. First, we compared our approach with existing methods and with our FPSS algorithm using *PhoneticMaps*, and we described the experiment where the proposed methods was applied and showed the results obtained using Brazilian Portuguese words. Finally, we presented the results from an experiment where we applied our solution in a use case for finding drug names and temporal tokens with

Table 4.12: Drugs with the highest number of spelling errors.

Drug	Similar words	TP	FP	FN	Precision	Recall	F1
propanolol	52	48	2	1	0.9600	0.9796	0.9697
glibenclamida	49	48	0	0	1.0000	1.0000	1.0000
anlodipino	49	42	4	1	0.9130	0.9767	0.9438
medroxiprogesterona	47	43	0	4	1.0000	0.9149	0.9556
metoclopramida	46	44	0	1	1.0000	0.9778	0.9888
loratadina	46	36	7	2	0.8372	0.9474	0.8889
dexametasona	45	36	0	9	1.0000	0.8000	0.8889
furosemida	43	26	1	0	0.9630	1.0000	0.9811
prednisona	42	29	0	4	1.0000	0.8788	0.9355
hidroclorotiazida	41	40	0	1	1.0000	0.9756	0.9877
diclofenaco	41	36	3	2	0.9231	0.9474	0.9351
ciprofloxacino	37	34	0	3	1.0000	0.9189	0.9577
espironolactona	36	36	0	0	1.0000	1.0000	1.0000
salbutamol	36	35	0	1	1.0000	0.9722	0.9859
clonazepam	34	34	0	0	1.0000	1.0000	1.0000
beclometasona	33	32	0	1	1.0000	0.9697	0.9846
dexclorfeniramina	31	28	0	3	1.0000	0.9032	0.9492
metronidazol	30	28	1	1	0.9655	0.9655	0.9655
prednisolona	30	28	1	1	0.9655	0.9655	0.9655
isossorbida	29	26	1	0	0.9630	1.0000	0.9811
TOTAL	797	709	20	35	0.9726	0.9530	0.9627

misspelling errors in a set of more than 4 thousand medical records, showing this structure is well adapted for calculating string and phonetic similarity between misspelt words.

Table 4.13: Temporal tokens and misspelt variations.

Temporal token	Misspelt occurrences	Examples
atualmente	21	altualmente – autualmente
anteriormente	17	antariormente – ateriormente
trimestre	16	timestre – trrimestre
madrugada	14	amadrugada – matrugada
recentemente	12	receentemente – recntemente
minuto	12	minhutos – mninutos
fevereiro	11	defevereiro – fevreiro
novembro	9	denovembro – novmebro
semana	9	semanada – semqanas
anterior	8	annterior – eanterior
recente	7	reacente – rescentes
dezembro	6	desembro – edzembro
outubro	6	outrubro – oututbro
setembro	5	setembor – setmebro
quinzenal	4	qeuizenal – qunzenal
domingo	3	domingoe – dormingo
janeiro	1	jamneiro

Chapter 5

Conclusions

Information Extraction (IE) has emerged as a text mining technique to identify specific information within unstructured data sources, as textual documents, making the information more suitable for information processing tasks. The understanding of temporal expressions extracted from text is fundamental for language understanding and an important sub-task for language processing applications, making it possible to identify and position extracted events in a chronological order. However, temporal information cannot be always accurately described, and imprecise temporal expressions should be handled – imprecise temporal expressions can reach up to 35% of the amount of temporal data in specific domains, such as in clinical narratives. Additionally, the problem of identifying temporal expressions in the text can still be more complicated when we consider that such expressions might also carry misspellings.

5.1 Contributions

In this thesis, we addressed the overall problem of dealing with imprecise temporal expressions within the IE process. Due to the lack in the current annotation standards regarding to how to normalise imprecise timexes in terms of values, it is not possible the use temporal-related logics and arithmetic to manipulate such inaccurate information. We proposed a normalisation methodology to capture the way people understand and reason about the vagueness carried by such kind of imprecise temporal data, and we addressed the impact of spelling errors when trying to identify different concepts within the source text, including temporal expressions.

The main contribution in this thesis is a methodology for the normalisation of imprecise temporal expressions extracted from text. Our methodology comprises different steps, from creating a set of questionnaires used to capture how people interpret vague descriptions of time in text. Answers were used as input data, from which we created histogram and fuzzy membership functions (MSF) during the pre-processing step. Then, we applied statistical regression and MLP techniques in order to evaluate which would be the most suitable model for each kind of imprecise temporal expression being evaluated. We used F1-score to calculate how similar two membership functions are, and to choose the suitable representation model for each kind of imprecise temporal expression. We also proposed a weighted F1-score variation ($F1_{3D}$) in order to identify where the differences are when comparing two membership functions. We applied the proposed methodology for three kinds of imprecise timexes, but the approach could be used for all of the kinds of imprecise temporal expression that were presented.

The proposed methodology produced normalisation models which were able to capture the vagueness carried by certain imprecise temporal expressions. For example, the Linear Regression Log(A) model resulted for the MV expressions in the form “less than N <granularity>"

in Portuguese, resulted (23,65,85,96) as the parameters that define the trapezoidal membership function for the expression “less than 90 days". Curiously, the resulted definition includes, even with a small confidence level, some values that are greater than 90 (91 to 95, as the last parameter 96 has confidence equals 0). That led us to believe temporal imprecision is not mathematically reasoned, but there is a level of uncertainty that is able to cross the boundary limits defined by the numerical values found within the temporal expressions.

Additionally, when comparing MV and IV imprecise temporal expressions between Portuguese and English languages, we can observe the way people understand that same kind of expression in two different languages is only about 75% similar. We can assign that discrepancy to different aspects, including the way questionnaires were design, or using different domains to write the sentences used in each questionnaire for each language. However, such divergence can also occur due to the differences that can be observed amongst people from different cultures.

We also reached additional contributions regarding to temporal information extraction and dealing with spelling errors within the IE process, as listed below.

We developed two approaches for time expression identification, as used in the SemEval-2015 Task 6, Clinical TempEval: a rule-based system that favoured recall and a machine learning approach built using readily available components, which was able to achieve a competitive F1 performance in a short development time. We discussed how they perform relative to each other, and how characteristics of the corpus affect outcomes and the suitability of the two approaches.

We suggested that inconsistent data, such as those found in the Clinical TempEval corpus, tend to lower the precision of rule-based systems. Thus, the appearance of a superior result by our machine learning system is therefore not to be taken at face value, as the machine learning system may have learned regularities in an incorrect annotation style, rather than having learned to accurately find time expressions. Machine learning systems have a flexibility and power in finding non-obvious cues to more subtle patterns, which makes them successful in linguistically complex tasks, but also gives them a deceptive appearance of success where the irregularity in a task comes not from its inherent complexity but from flaws in the dataset.

Adapting annotation guidelines of temporal semantics to clinical notes is a significant and challenging task. Thus, we examined temporal expressions in the first major corpus released under this standard, in order to make the findings of this data-driven analysis could be used as recommendations be considered in future manual annotation efforts. We investigated where the standard has proven difficult to apply, and gave a series of recommendations regarding temporal annotation. We also detailed the results of a principled analysis of expert manual annotations of temporal expressions in the THYME schema over a corpus of clinical notes. Discrepancies between annotations and the guidelines were found in multiple categories: a) the spans or temporal expressions were not always correct; b) ambiguity remained regarding the correct timex class, as happened also in TimeML; c) wording in the guidelines was sometimes misinterpreted leading to non-markable timexes being annotated; and d) some confusion appeared around the annotation of complex SET-type timexes and their quantifiers.

We presented an approach of fast phonetic similarity search coupled with an extended version of the Wordnet repository to support this structure. The most important difference of our approach is the utilisation of an indexed structure that stores phonetic information to be used by a novel string similarity search algorithm. Our approach has three main contributions: a) an indexed data structure (PhoneticMap), b) a novel string similarity algorithm (StringSim), and c) we integrated the previous contributions with Princeton WordNet (PWN) to implement the fast phonetic similarity search. $StringSim$ is based on the notion of penalty, keeping the similarity values higher for words with less than 4–6 differences. In contrast, it decreases and converges the similarity values to zero faster comparing to other known string similarity metrics. We performed

a set of extensive experiments in order to validate our approach. The proposed similarity search algorithm has good precision results and it executes faster than one version of the algorithm that does not use the $PhoneticMap$, when using Brazilian Portuguese words. Finally, we applied our solution to a case study for discovering misspelt forms of drug names and timexes in a set of medical records. Fast phonetic similarity search has proven to be well adapted for combining string and phonetic similarity when finding misspelt words.

5.2 Future Work

Temporal Information Extraction

The rule-based system we developed for the Clinical TempEval task favoured recall. However, inconsistent data found in the Clinical TempEval corpus tended to lower the system precision. We plan to analyse new datasets in Portuguese and English to improve the set of rules used to identify timexes, in order to find patterns that correspond to more complex expressions.

Based on the recommendations we ventured to make to be considered in future manual annotation efforts, we plan to join different temporal annotation standards into a single document in order to avoid annotators having to piece together several guidelines to figure out what to annotate, including improvements in each annotation rule description, and including more positive and negative examples. We also plan to evaluate whether or not using a high recall rule-based system to prepare a corpus, creating annotations to be reviewed by human annotators, would improve the quality of a gold standard corpus. We believe for those annotations that can be defined by simple, unambiguous rules, and where this is the case, the rules will most likely outperform the human annotator in terms of recall, and the tendency of a poor correction of missing spans, would be outweighed by the increased number of annotations found. We also plan to adapt the annotation guidelines to include additional features to describe fuzzy values for imprecise temporal expressions. Based on the experiments performed, we developed normalisation models for imprecise temporal expression in English and Portuguese. We used F1-score to evaluate how similar two languages are when comparing the same kind of imprecise timex. Similarity between English and Portuguese reached about 75% for MV and IV expressions, and about 40% for PR expressions. Because the questionnaires designed were developed with different domains for each language, it is not possible to clearly state whether the reason for such differences is due to cultural diversity, or just a casualty. An experiment focused on measuring that similarity has to be performed, using a questionnaire comprising questions specifically designed to calculate similarity between languages in terms of imprecise reasoning. Furthermore, different domains can affect the way people reason about imprecision, and a comparison using similar questions and sentences applied to distinct domains can show whether the perception of temporal imprecision is domain dependant.

We evaluated the final normalisation models for imprecise timexes using a variation of F1 that uses the area of MSFs as input to measure how similar two different MSFs are, and the proposed $F1_{3D}$-score as a complementary metric to describe whether the differences between two MSFs are more concentrated in the top (confidence=1) or in the bottom (confidence=0). As complementary metrics, F1 and $F1_{3D}$ work together to describe the similarity between two MSFs or two imprecise models. We plan to perform further experiments in order to validate both metrics and formalise the correct way to apply, represent and interpret their meanings.

The normalisation of imprecise temporal expressions in terms of the `value` attribute can improve the amount of extracted events connected to the timeline. Regardless of representing up to 35% of the amount of temporal data in specific domains, we plan to perform further

search-based experiments over extracted events from medical records in order to evaluate what is the relevance of dealing with such imprecise temporal data along the IE process in that specific domain.

Similarity Search

As a language-dependent and modular approach, the *PhoneticMap* can be encoded differently, without affecting the other components of an Information Extraction system. The approach is tailored to adapt phonetic match to be used over large repositories and it is adaptable to different languages, such as English or Spanish, to create new instances for the $PhoneticMap$ structure, and the $PhoneticMapSim$ and $PhoneticSearch$ function. We plan to work on finding a version of the English PhoneticMap that can produce acceptable time responses. We have been performing tests with Soundex and Double-methaphone functions to create an English version of a PhoneticMap. However, the way we indexed the PhoneticMap entries did not produce satisfactory results in terms of time response during phonetic searches.

We also want to integrate our solution in a framework for a Medical Record Information Extraction System to address the problem of dealing with spelling errors within the extraction process. We plan to explore the usage of alternative methods to optimally tune the parameters involved in the proposed hybrid similarity metrics, and to compare our proposed solution with other phonetic search approaches.

When trying to apply the phonetic and string similarity thresholds defined for misspelt drug names described in Table 4.11 to select the possible misspelt temporal tokens, we observed a considerable number of false positives and false negatives. That led us to perform a manual selection of correct misspelt words for each temporal token, discharging those false positives. We want to explore different threshold values to be used for different string length ranges in order to whether the threshold parameters can be more efficient when they can be specifically set for different string sizes.

Bibliography

[Abderrahim et al., 2013] Abderrahim, M. A., Abderrahim, M. E. A., and Chikh, M. A. (2013). Using arabic wordnet for semantic indexation in information retrieval system. *CoRR*, abs/1306.2499.

[Ahn et al., 2005] Ahn, D., Adafre, S. F., and Rijke, M. (2005). Extracting temporal information from open domain text: A comparative exploration. *Journal of Digital Information Management*, 3:2005.

[Allen, 1983] Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

[Allison and Dix, 1986] Allison, L. and Dix, T. I. (1986). A bit-string longest-common-subsequence algorithm. *Inf. Process. Lett.*, 23(6):305–310.

[Alonso et al., 2007] Alonso, O., Gertz, M., and Baeza-Yates, R. (2007). On the value of temporal information in information retrieval. *SIGIR Forum*, 41(2):35–41.

[Álvarez et al., 2007] Álvarez, M., Pan, A., Raposo, J., Bellas, F., and Cacheda, F. (2007). Using clustering and edit distance techniques for automatic web data extraction. In *Web Information Systems Engineering–WISE 2007*, pages 212–224. Springer.

[Anantharangachar et al., 2013] Anantharangachar, R., Ramani, S., and Rajagopalan, S. (2013). Ontology guided information extraction from unstructured text. *CoRR*, abs/1302.1335.

[Aranha, 2007] Aranha, C. N. (2007). *Uma abordagem de pré-processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil.

[Ashish et al., 2009] Ashish, N., Mehrotra, S., and Pirzadeh, P. (2009). Xar: An integrated framework for information extraction. In *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering - Volume 04*, CSIE '09, pages 462–466, Washington, DC, USA. IEEE Computer Society.

[Bartak et al., 2013] Bartak, R., Morris, R., and Venable, K. (2013). *An Introduction to Constraint-Based Temporal Reasoning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

[Bassara, 2007] Bassara, A. (2007). *Temporalizing Ontology*, page 211;290. Springer Verlag, Dordrecht.

[Batsakis and Petrakis, 2011] Batsakis, S. and Petrakis, E. (2011). Representing temporal knowledge in the semantic web: The extended 4d fluents approach. In Hatzilygeroudis, I. and Prentzas, J., editors, *Combinations of Intelligent Methods and Applications*, volume 8 of *Smart Innovation, Systems and Technologies*, pages 55–69. Springer Berlin Heidelberg.

[Bethard, 2013] Bethard, S. (2013). Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14, Atlanta, Georgia, USA. Association for Computational Linguistics.

[Bethard et al., 2015] Bethard, S., Derczynski, L., Pustejovsky, J., and Verhagen, M. (2015). SemEval-2015 Task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics.

[Bethard et al., 2007] Bethard, S., Martin, J. H., and Klingenstein, S. (2007). Timelines from text: Identification of syntactic temporal relations. In *ICSC*, pages 11–18. IEEE Computer Society.

[Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing.

[Blamey et al., 2013] Blamey, B., Crick, T., and Oatley, G. (2013). 'the first day of summer': Parsing temporal expressions with distributed semantics. In Bramer, M. and Petridis, M., editors, *Research and Development in Intelligent Systems XXX*, pages 389–402. Springer International Publishing.

[Blaylock et al., 2011] Blaylock, N., de Beaumont, W., Allen, J., and Jung, H. (2011). Towards an owl-based framework for extracting information from clinical texts. In *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '11, pages 636–640, New York, NY, USA. ACM.

[Bocek et al., 2008] Bocek, T., Hunt, E., Hausheer, D., and Stiller, B. (2008). Fast similarity search in peer-to-peer networks. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pages 240–247. IEEE.

[Bocek et al., 2007] Bocek, T., Hunt, E., Stiller, B., and Hecht, F. (2007). Fast similarity search in large dictionaries. Technical Report ifi-2007.02, Department of Informatics, University of Zurich. http://fastss.csg.uzh.ch/.

[Bocek et al., 2009] Bocek, T., Victora Hecht, F., Hausheer, D., Hunt, E., and Stiller, B. (2009). Mobile p2p fast similarity search. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–2. IEEE.

[Boguraev and Ando, 2007] Boguraev, B. and Ando, R. K. (2007). Effective use of TimeBank for TimeML analysis. In *Annotating, extracting and reasoning about time and events*, pages 41–58. Springer.

[Bona, 2002] Bona, C. (2002). Avaliação de processos de software: Um estudo de caso em xp e iconix. Master's thesis, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina (UFSC).

[Burman et al., 2011] Burman, A., Jayapal, A., Kannan, S., Kavilikatta, M., Alhelbawy, A., Derczynski, L., and Gaizauskas, R. (2011). USFD at KBP 2011: Entity linking, slot filling and temporal bounding. In *Proceedings of the Text Analysis Conference*.

[Cardoso et al., 2011] Cardoso, P. C., Maziero, E. G., Jorge, M. L. C., Seno, E. R., Di Felippo, A., Rino, L. H. M., Nunes, M. d. G. V., and Pardo, T. A. S. (2011). Cstnews – a discourse-annotated

corpus for single and multi-document summarization of news texts in brazilian portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105, Cuiabá, Brazil.

[Caselli, 2009] Caselli, T. (2009). *Time, Events and Temporal Relations: an Empirical Model for Temporal Processing of Italian Texts*. PhD thesis, Università di Pisa, Pisa, Italy.

[Catarino, 1999] Catarino, D. (1999). Gramatica on-line. `http://www.gramaticaonline.com.br/`. Accessed: Jul, 2013.

[Chambers, 2013] Chambers, N. (2013). Navytime: Event and time ordering from raw text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 73–77, Atlanta, Georgia, USA. Association for Computational Linguistics.

[Chang and Manning, 2012] Chang, A. X. and Manning, C. D. (2012). Sutime: A library for recognizing and normalizing time expressions. In Calzolari, N., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *LREC*, pages 3735–3740. European Language Resources Association (ELRA).

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3).

[Coelho and Raposo, 2005] Coelho, A. L. V. and Raposo, A. B. (2005). Dealing with imprecision in temporal interdependencies between collaborative tasks: A fuzzy perspective.

[Cohen et al., 2003] Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. pages 73–78.

[Corney et al., 2008] Corney, D., Byrne, E., Buxton, B., and Jones, D. (2008). A logical framework for template creation and information extraction. In Lin, T., Xie, Y., Wasilewska, A., and Liau, C.-J., editors, *Data Mining: Foundations and Practice*, volume 118 of *Studies in Computational Intelligence*, pages 79–108. Springer Berlin Heidelberg.

[Costa and Branco, 2012] Costa, F. and Branco, A. (2012). Extracting temporal information from portuguese texts. In de Medeiros Caseli, H., Villavicencio, A., Teixeira, A. J. S., and Perdigão, F., editors, *PROPOR*, volume 7243 of *Lecture Notes in Computer Science*, pages 99–105. Springer.

[Cunningham et al., 2011] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*. The University of Sheffield.

[Cunningham et al., 2000] Cunningham, H., Maynard, D., and Tablan, V. (2000). JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield.

[Dantchev, 2013] Dantchev, M. (2013). Wordnet 2.1 overview. `https://pipl.com/directory/name/Dantchev/Marin/`. Accessed: Aug, 2013.

[Davis and Goadrich, 2006] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240, New York, NY, USA. ACM.

[Derczynski, 2013] Derczynski, L. (2013). *Determining the Types of Temporal Relations in Discourse*. PhD thesis, University of Sheffield, UK.

[Derczynski et al., 2015] Derczynski, L., Strötgen, J., Campos, R., and Alonso, O. (2015). Time and information retrieval: Introduction to the special issue. *Information Processing & Management*, 51(6):786 – 790.

[Droppo and Acero, 2010] Droppo, J. and Acero, A. (2010). Context dependent phonetic string edit distance for automatic speech recognition. In *ICASSP*, pages 4358–4361. IEEE.

[Fagerberg, 2014] Fagerberg, A. (2014). Temporal information extraction using regular expressions. `http://www.antonfagerberg.com/files/tempex_anton_fagerberg.pdf`. Accessed: Mar, 2014.

[Fenz et al., 2012a] Fenz, D., Lange, D., Rheinländer, A., Naumann, F., and Leser, U. (2012a). Efficient similarity search in very large string sets. In *Proceedings of the 24th international conference on Scientific and Statistical Database Management*, SSDBM'12, pages 262–279, Berlin, Heidelberg. Springer-Verlag.

[Fenz et al., 2012b] Fenz, D., Lange, D., Rheinländer, A., Naumann, F., and Leser, U. (2012b). Efficient similarity search in very large string sets. In Ailamaki, A. and Bowers, S., editors, *Scientific and Statistical Database Management*, volume 7338 of *Lecture Notes in Computer Science*, pages 262–279. Springer Berlin Heidelberg.

[Ferreira, 2004] Ferreira, A. B. (2004). *Novo dicionário Aurélio da língua portuguesa*. Positivo.

[Ferro et al., 2005] Ferro, L., Gerber, L., Mani, I., Sundheim, B., and Wilson, G. (2005). TIDES 2005 standard for the annotation of temporal expressions. Technical report, The MITRE Corporation.

[Filannino and Nenadic, 2014] Filannino, M. and Nenadic, G. (2014). Mining temporal footprints from Wikipedia. In *Proceedings of the First AHA! workshop*, pages 7–13.

[Fleiss et al., 1981] Fleiss, J. L., Levin, B., and Paik, M. C. (1981). The Measurement of Interrater Agreement. pages 212–236.

[Frozza and dos Santos Mello, ez06] Frozza, A. A. and dos Santos Mello, R. (20varez06). Um método para determinar a equivalência semãntica entre esquemas gml. In *GeoInfo*, pages 283–294.

[Gardner and Dorling, 1998] Gardner, M. W. and Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14-15):2627–2636.

[Ghawi and Cullot, 2007] Ghawi, R. and Cullot, N. (2007). Database-to-ontology mapping generation for semantic interoperability. In *Third International Workshop on Database Interoperability (InterDB 2007)*.

[Godbole et al., 2010] Godbole, S., Bhattacharya, I., Gupta, A., and Verma, A. (2010). Building re-usable dictionary repositories for real-world text mining. In Huang, J., Koudas, N., Jones, G. J. F., Wu, X., Collins-Thompson, K., and An, A., editors, *CIKM*, pages 1189–1198. ACM.

[Gomaa and Fahmy, 2013] Gomaa, W. H. and Fahmy, A. A. (2013). Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18. Full text available.

[Gomes et al., 2004] Gomes, P., Pereira, F. C., Paiva, P., Seco, N., Carreiro, P., Ferreira, J. L., and Bento, C. (2004). Using wordnet for case-based retrieval of uml models. *AI Commun.*, 17(1):13–23.

[Gonzalez et al., 2012] Gonzalez, A., Laparra, E., and Rigau, G. (2012). Multilingual central repository version 3.0: upgrading a very large lexical knowledge base. In *6th Global WordNetConference*, Matsue, Japan.

[Gooch, 2012] Gooch, P. (2012). *A modular, open-source information extraction framework for identifying clinical concepts and processes of care in clinical narratives*. PhD thesis, City University London, London, UK.

[Goralwalla et al., 2001] Goralwalla, I., Leontiev, Y., Ozsu, M., Szafron, D., and Combi, C. (2001). Temporal granularity: Completing the puzzle. *Journal of Intelligent Information Systems*, 16:41–63.

[Gruber, 1993] Gruber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands. Kluwer Academic Publishers.

[Hall and Dowling, 1980] Hall, P. A. V. and Dowling, G. R. (1980). Approximate string matching. *ACM Comput. Surv.*, 12(4):381–402.

[Hamming, 1950] Hamming, R. (1950). Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160.

[Heeringa, 2004] Heeringa, W. (2004). *Measuring Dialect Pronunciation Differences Using Levenshtein Distance*. PhD thesis, University of Groningen, Groningen, Netherlands.

[Hjorland and Albrechtsen, 1995] Hjorland, B. and Albrechtsen, H. (1995). Toward a new horizon in information science: Domain-analysis. *JASIS*, 46(6):400–425.

[Hovy et al., 2014] Hovy, D., Plank, B., and Søgaard, A. (2014). When POS data sets don't add up: Combatting sample bias. In *Proc. LREC*, pages 4472–4475. LREC.

[ISO, 2007] ISO (2007). Language resource management — semantic annotation framework (semaf) — part 1: Time and events. In *ISO/TC37/SC 4N269 rev04*. ISO Report.

[Jellouli and Mohajir, 2011] Jellouli, I. and Mohajir, M. (2011). An ontology-based approach for web information extraction. In *Information Science and Technology (CIST), 2011 Colloquium in*, pages 5–5.

[Khabsa et al., 2012] Khabsa, M., Treeratpituk, P., and Giles, C. L. (2012). Ackseer: a repository and search engine for automatically extracted acknowledgments from digital libraries. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 185–194. ACM.

[Kilgarriff, 2010] Kilgarriff, A. (2010). Dante: A detailed, accurate, extensive, available english lexical database. North American Association for Computational Linguistics (NAACL-HLT).

[Kolomiyets, 2012] Kolomiyets, O. (2012). *Algorithms for Temporal Information Processing of Text and their Applications*. PhD thesis, Informatics Section, Department of Computer Science, Faculty of Engineering Science. Moens, Marie-Francine and De Schreye, Daniel (supervisors).

[Kolomiyets and Moens, 2010] Kolomiyets, O. and Moens, M.-F. (2010). KUL: recognition and normalization of temporal expressions. In *Proceedings of SemEval-2 5th Workshop on Semantic Evaluation - ACL SigLex,*, pages 325–328. ACL.

[Kolomiyets and Moens, 2013] Kolomiyets, O. and Moens, M.-F. (2013). KUL: A data-driven approach to temporal parsing of documents. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013),*, pages 83–87. ACL.

[Ladefoged and Maddieson, 1996] Ladefoged, P. and Maddieson, I. (1996). *The sounds of the world's languages*. Blackwell, Oxford, UK.

[Latha et al., 2007] Latha, K., Kalimuthu, S., and Dr.Rajaram, R. (2007). Information extraction from biomedical literature using text mining framework. *International Journal of Imaging Science and Engineering (IJISE)*, 1(1).

[Lee and Katz, 2009] Lee, C. M. and Katz, G. (2009). Error analysis of the tempeval temporal relation identification task. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 138–145. ACL.

[Leenoi et al., 2009] Leenoi, D., Supnithi, T., and Aroonmanakun, W. (2009). Building thai wordnet with a bi-directional translation method. In Zhang, M., Li, H., Lua, K.-T., and Dong, M., editors, *IALP*, pages 48–52. IEEE Computer Society.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

[Li and Shen, 2009] Li, W. and Shen, N. (2009). Ontology-based drug product information extraction system. In *BMEI*, pages 1–4. IEEE.

[Li et al., 2005] Li, Y., Bontcheva, K., and Cunningham, H. (2005). SVM Based Learning System For Information Extraction. In Winkler, J., Niranjan, M., and Lawerence, N., editors, *Deterministic and Statistical Methods in Machine Learning: First International Workshop, 7–10 September, 2004*, volume 3635 of *Lecture Notes in Computer Science*, pages 319–339, Sheffield, UK. Springer.

[Li et al., 2009] Li, Y., Bontcheva, K., and Cunningham, H. (2009). Adapting SVM for Data Sparseness and Imbalance: A Case Study on Information Extraction. *Natural Language Engineering*, 15(2):241–271.

[Lin and Och, 2004] Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 605. Association for Computational Linguistics.

[Ling and Weld, 2010] Ling, X. and Weld, D. S. (2010). Temporal information extraction. In Fox, M. and Poole, D., editors, *AAAI*. AAAI Press.

[Llorens et al., 2012] Llorens, H., Derczynski, L., Gaizauskas, R. J., and Saquete, E. (2012). TIMEN: An open temporal expression normalisation resource. In *LREC*, pages 3044–3051. ELRA.

[Lucrédio et al., 2012] Lucrédio, D., M. Fortes, R. P., and Whittle, J. (2012). Moogle: a metamodel-based model search engine. *Softw. Syst. Model.*, 11(2):183–208.

[Maedche et al., 2002] Maedche, A., Neumann, G., and Staab, S. (2002). Bootstrapping an ontology-based information extraction system. In *Studies in Fuzziness and Soft Computing, Intelligent Exploration of the Web*. Springer.

[Mani, 2003] Mani, I. (2003). Recent developments in temporal information extraction. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, *RANLP*, volume 260 of *Current Issues in Linguistic Theory (CILT)*, pages 45–60. John Benjamins, Amsterdam/Philadelphia.

[Mani et al., 2004] Mani, I., Pustejovsky, J., and Sundheim, B. (2004). Introduction to the special issue on temporal information processing. 3(1):1–10.

[Mann, 1986] Mann, V. A. (1986). Distinguishing universal and language-dependent levels of speech perception: Evidence from japanese listeners' perception of english. *Cognition*, 24(3):169 – 196.

[Maslennikov and Chua, 2007] Maslennikov, M. and Chua, T.-S. (2007). A multi-resolution framework for information extraction from free text. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL*. The Association for Computational Linguistics.

[Mazur and Dale, 2010] Mazur, P. and Dale, R. (2010). Wikiwars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 913–922, Stroudsburg, PA, USA. Association for Computational Linguistics.

[McCallum, 2006] McCallum, A. (2006). Part-of-speech tagging & hidden markov model intro. In *Computational Linguistics CMPSCI 591N*.

[Meystre et al., 2008] Meystre, S., Savova, G., Kipper-Schuler, K., and Hurdle, J. (2008). Extracting information from textual documents in the electronic health record: A review of recent research. *IMIA Yearbook of Medical Informatics*, pages 128–144.

[Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

[Moens, 2006] Moens, M.-F. (2006). *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Monge and Elkan, 1996] Monge, A. and Elkan, C. (1996). The field matching problem: Algorithms and applications. In *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 267–270.

[Motta and Osborne, 2012] Motta, E. and Osborne, F. (2012). Making sense of research with rexplore. In Glimm, B. and Huynh, D., editors, *International Semantic Web Conference (Posters & Demos)*, volume 914 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Muller et al., 2004] Muller, H. M., Kenny, E. E., and Sternberg, P. W. (2004). Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biol*, 2:309.

[Nagypál and Motik, 2003] Nagypál, G. and Motik, B. (2003). A fuzzy model for representing uncertain, subjective and vague temporal knowledge in ontologies. In *In Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics, (ODBASE), volume 2888 of LNCS*, pages 906–923. Springer.

[Navigli, 2009] Navigli, R. (2009). Word sense disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

[Nedellec and Nazarenko, 2006] Nedellec, C. and Nazarenko, A. (2006). Ontologies and information extraction. *CoRR*, abs/cs/0609137.

[Osborne et al., 2013] Osborne, F., Motta, E., and Mulholland, P. (2013). Exploring scholarly data with rexplore. In *International Semantic Web Conference (1)*, pages 460–477.

[Paterson and Dancik, 1994] Paterson, M. and Dancik, V. (1994). Longest common subsequences. In *In Proc. of 19th MFCS, number 841 in LNCS*, pages 127–142. Springer.

[Pavel and Euzenat, 2011] Pavel, S. and Euzenat, J. (2011). Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, PP(99).

[Pedrycz and Gomide, 1998] Pedrycz, W. and Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. Complex adaptive systems. NetLibrary, Incorporated.

[Pustejovsky, 2006] Pustejovsky, J. (2006). Unifying linguistic annotations: A TimeML case study. In *Proceedings of Text, Speech, and Dialogue Conference*.

[Pustejovsky et al., 2003a] Pustejovsky, J., Castano, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., and Katz, G. (2003a). TimeML: Robust specification of event and temporal expressions in text. In *in Fifth International Workshop on Computational Semantics (IWCS-5)*.

[Pustejovsky et al., 2003b] Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al. (2003b). The TimeBank corpus. In *Proceedings of the Corpus Linguistics Conference*, volume 2003, page 40.

[Pustejovsky et al., 2003c] Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., and Lazo, M. (2003c). The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, Lancaster.

[Pustejovsky et al., 2010] Pustejovsky, J., Lee, K., Bunt, H., and Romary, L. (2010). ISO-TimeML: An international standard for semantic annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. ELRA.

[Pustejovsky and Moszkowicz, 2012] Pustejovsky, J. and Moszkowicz, J. (2012). The role of model testing in standards development: The case of ISO-Space. In *LREC*, pages 3060–3063.

[Pustejovsky and Stubbs, 2012] Pustejovsky, J. and Stubbs, A. (2012). *Natural language annotation for machine learning*. O'Reilly Media, Inc.

[Roberts et al., 2009] Roberts, A., Gaizauskas, R. J., Hepple, M., Demetriou, G., Guo, Y., Roberts, I., and Setzer, A. (2009). Building a semantically annotated corpus of clinical texts. *Journal of Biomedical Informatics*, 42(5):950–966.

[Saggion et al., 2007] Saggion, H., Funk, A., Maynard, D., and Bontcheva, K. (2007). Ontology-based information extraction for business intelligence. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 843–856, Berlin, Heidelberg. Springer-Verlag.

[Sanampudi and Kumari, 2010] Sanampudi, S. K. and Kumari, G. (2010). Article: Temporal reasoning in natural language processing: A survey. *International Journal of Computer Applications*, 1(4):53–57. Published By Foundation of Computer Science.

[Sauri et al., 2006] Sauri, R., Littman, J., Gaizauskas, R., Setzer, A., and Pustejovsky, J. (2006). TimeML Annotation Guidelines, Version 1.2.1.

[Schilder and Habel, 2003] Schilder, F. and Habel, C. (2003). Temporal Information extraction for temporal question answering. In *Proceedings of the 2003 AAAI Spring Symposium in New Directions in Question Answering*, Stanford University, Palo Alto, USA.

[Schockaert, 2005] Schockaert, S. (2005). Construction of membership functions for fuzzy time periods. In *Proceedings of the ESSLLI 2005 Student Session*.

[Schockaert et al., 2008] Schockaert, S., Cock, M. D., and Kerre, E. E. (2008). Fuzzifying Allen's temporal interval relations. *IEEE T. Fuzzy Systems*, 16(2):517–533.

[Senger et al., 2010] Senger, C., Kaltschmidt, J., Schmitt, S. P. W., Pruszydlo, M. G., and Haefeli, W. E. (2010). Misspellings in drug information system queries: Characteristics of drug name spelling errors and strategies for their prevention. *I. J. Medical Informatics*, 79(12):832–839.

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

[Stewart et al., 2009] Stewart, R., Soremekun, M., Perera, G., Broadbent, M., Callard, F., Denis, M., Hotopf, M., Thornicroft, G., and Lovestone, S. (2009). The South London and Maudsley NHS Foundation Trust Biomedical Research Centre (SLAM BRC) case register: development and descriptive data. *BMC Psychiatry*, 9:51.

[Strötgen et al., 2013] Strötgen, J., Zell, J., and Gertz, M. (2013). Heideltime: Tuning english and developing spanish resources for tempeval-3. In *2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7h International Workshop on Semantic Evaluation (SemEval 2013)*, pages 15–19, Atlanta, Georgia, USA. ACL.

[Stvilia, 2007] Stvilia, B. (2007). A model for ontology quality evaluation. *First Monday*, 12(12).

[Styler et al., 2014] Styler, W., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P., Erickson, B., Miller, T., Lin, C., Savova, G., and Pustejovsky, J. (2014). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

[Su et al., 2008] Su, Z., Ahn, B.-R., Eom, K.-Y., Kang, M.-K., Kim, J.-P., and Kim, M.-K. (2008). Plagiarism detection using the levenshtein distance and smith-waterman algorithm. In *Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on*, pages 569–569.

[Sun et al., 2013] Sun, W., Rumshisky, A., and Uzuner, O. (2013). Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

[Tissot et al., 2015a] Tissot, H., Gorrell, G., Roberts, A., Derczynski, L., and Fabro, M. D. D. (2015a). UFPRSheffield: Contrasting rule-based and support vector machine approaches to time expression identification in clinical tempeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 835–839, Denver, Colorado. Association for Computational Linguistics.

[Tissot et al., 2014] Tissot, H., Peschl, G., and Fabro, M. D. D. (2014). Fast phonetic similarity search over large repositories. In *Database and Expert Systems Applications - 25th International Conference, DEXA 2014, Munich, Germany, September 1-4, 2014. Proceedings, Part II*, pages 74–81.

[Tissot et al., 2015b] Tissot, H., Roberts, A., Derczynski, L., Gorrell, G., and Didonet Del Fabro, M. (2015b). Analysis of temporal expressions annotated in clinical notes. In *Proceedings of 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 93–102, London, UK. ACL.

[UzZaman and Allen, 2010] UzZaman, N. and Allen, J. F. (2010). Trips and trios system for tempeval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 276–283, Stroudsburg, PA, USA. Association for Computational Linguistics.

[UzZaman et al., 2013a] UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. (2013a). SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9. ACL.

[UzZaman et al., 2013b] UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J. F., and Pustejovsky, J. (2013b). SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the 7th International Workshop on Semantic Evaluations*.

[Velupillai et al., 2015] Velupillai, S., Mowery, D., South, B. R., Kvist, M., and Dalianis, H. (2015). Recent advances in clinical natural language processing in support of semantic analysis. *IMIA Yearbook of Medical Informatics*, pages 183–193.

[Verhagen, 2004] Verhagen, M. (2004). *Times Between the Lines: Embedding a Temporal Closure Component in a Mixed-initiative Temporal Annotation Framework*. PhD thesis, Waltham, MA, USA. AAI3148969.

[Verhagen et al., 2009] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., and Pustejovsky, J. (2009). The TempEval challenge: identifying temporal relations in text. *Language Resources and Evaluation*, 43(2):161–179.

[Verhagen et al., 2010] Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010). Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62. ACL.

[Wakil, 2002] Wakil, M. E. (Feb. 2002). Introducing text mining. In *Proceedings of the 9th Scientific Conference for Information Systems and Information Technology*, [Poster].

[Wimalasuriya and Dou, 2010a] Wimalasuriya, D. C. and Dou, D. (2010a). Components for information extraction: ontology-based information extractors and generic platforms. In Huang, J., Koudas, N., Jones, G. J. F., Wu, X., Collins-Thompson, K., and An, A., editors, *CIKM*, pages 9–18. ACM.

[Wimalasuriya and Dou, 2010b] Wimalasuriya, D. C. and Dou, D. (2010b). Ontology-based information extraction: An introduction and a survey of current approaches. *J. Inf. Sci.*, 36(3):306–323.

[Winkler, 1990] Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359.

[Wong et al., 2005] Wong, K.-F., Xia, Y., Li, W., and Yuan, C. (2005). An overview of temporal information extraction. *Int. J. Comput. Proc. Oriental Lang.*, 18(2):137–152.

[Yankova et al., 2008] Yankova, M., Saggion, H., and Cunningham, H. (2008). A framework for identity resolution and merging for multi-source information extraction. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[Zadeh, 1994] Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, 37(3):77–84.

[Zhou et al., 2005] Zhou, L., Friedman, C., Parsons, S., and Hripcsak, G. (2005). System architecture for temporal information extraction, representation and reasoning in clinical narrative reports. pages 869–73.

[Zhou et al., 2006] Zhou, L., Parsons, S., and Hripcsak, G. (2006). Handling implicit and uncertain temporal information in medical text. *AMIA Annu Symp Proc*, page 1158.

[Zhou et al., 2011] Zhou, X., Li, H., Lu, X., and Duan, H. (2011). Temporal expression recognition and temporal relationship extraction from chinese narrative medical records. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 1–4.

[Zobel and Dart, 1996] Zobel, J. and Dart, P. (1996). Phonetic string matching: Lessons from information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 166–172, New York, NY, USA. ACM.

Appendix A

Confidentiality Agreement to Access Information Stored in the *InfoSaude* System

PREFEITURA DA CIDADE DE FLORIANÓPOLIS
SECRETARIA MUNICIPAL DE SAÚDE
GERÊNCIA DE TECNOLOGIA E MODERNIZAÇÃO

29 May 2012

To whom it may concern.

This state is provided to verify that HEGLER CORREA TISSOT has presented your PhD research proposal named "Imprecise Temporal Information Extraction from Medical Records". After being examined by this committee, his project was approved to have access granted to the information stored in our health system, in accordance with the current imposed confidentiality restrictions.

Marcos Aurélio Geremias
Monitoring Committee for Research Projects in Health
Phone: +55 (48) 9624.9543
Email: marcos.geremias@gmail.com

Marcos Aurélio Geremias
Gerência de Tecnologia e Modernização-GTEC
Matrícula 19.138-8 / SMS-PMF

Appendix B

A SVM Approach to Time Expression Identification in Clinical TempEval

This work was developed by Dr. Genevieve Gorrell[1] (The University of Sheffield, UK) as one of two approaches we subbmited to SemEval 2015 Clinical TempEval [Bethard et al., 2015].

GATE provides an integration of LibSVM [Chang and Lin, 2011] technology with some modifications and short cuts enabling effective rapid prototyping for the task of locating and classifying named entities. This was used to quickly achieve competitive results. An initial system was created in a few hours, and although a couple of days were spent trying parameter and feature variants, the initial results could not be improved. No development effort was required, the system being used as "off the shelf" software.

State of the art machine learning approaches to timex recognition often use sequence labelling (e.g. CRF) to find timex bounds [UzZaman et al., 2013b], then a use separate instance-based classification step (e.g. with SVM) to classify them [Sun et al., 2013]. Our approach uses SVM to implement separate named entity recognizers for each class, then makes a final selection for each span based on probability. GATE's LibSVM integration incorporates the uneven margins parameter (UM) [Li et al., 2009], which has been shown to improve results on imbalanced datasets especially for smaller corpora. In positioning the hyperplane further from the (smaller) positive set, we compensate for a tendency in smaller corpora for the larger (negative) class to push away the separator in a way that it does not tend to do when sufficient positive examples exist for them to populate their space more thoroughly, as this default behaviour can result in poor generalization and a conservative model.

Since NLP tasks such as NER often do involve imbalanced datasets, this inclusion, as well as robust default implementation choices for NLP tasks, makes it easy to get a respectable result quickly using GATE's SVM/UM, as our entry demonstrates.

The feature set used is:

- String of the current token plus the preceding and ensuing five.
- Part of speech of the current token, plus the preceding and ensuing five.
- If a date has been detected for this span using the Date Normalizer rule-based date detection and normalization resource in GATE, then the type of date in this location is included as a feature. The mere presence of such a date annotation may be the most important aspect of this feature. Note that this Date Normalizer was not used in HINX, where a bespoke solution was developed.

[1] `http://www.dcs.shef.ac.uk/~genevieve/`

- As above, but using the "complete" feature on the date, to indicate whether the date present in this location is a fully qualified date. This may be of value as an indicator of the quality of the rule-based date annotation.

A probabilistic polynomial SVM is used with an order of 3. Probabilistic SVMs allow us to apply confidence thresholds later, which further permits us: 1) to tune to the imbalanced dataset and task constraints, 2) to use the "one vs rest" method for transforming the multiclass problem to a set of binary problems, and 3) to select the final class for the time expression. In the "one vs rest" approach, one classifier is created for each class, allowing it to be separated from all others, and the class with the highest confidence score is chosen. An uneven margins parameter of 0.4 is selected on the basis of previous work [Li et al., 2005].

Two classifiers are trained for each class; one to identify the start of the entity and another to identify the end. This information is then post-processed into entity spans first by removing orphaned start or end tags and secondly by filtering out entities with lengths (in number of words) that did not appear in the training data. Finally, where multiple annotations overlap, a confidence score is used to select the strongest candidate. A separate confidence score is also used to remove weak entities.

Table B.1 shows negligible difference between a linear and polynomial SVM (degree 3). A confidence threshold of 0.25 was selected empirically. Task training data was split 50:50 to form training and test sets to produce these figures. An additional experiment involved including the output from the HINX rule-based system as features for the SVM. This did not improve the outcome.

Table B.1: SVM Tuning Results.

SVM	Threshold	P	R	F1
Linear	0.2	0.68	0.59	0.63
Linear	0.4	0.76	0.55	0.64
Poly (3)	0.2	0.64	0.61	0.63
Poly (3)	0.25	0.69	0.61	**0.65**
Inc. hinx feats	0.25	0.72	0.54	0.62

Appendix C

Sentences used in the Portuguese Questionnaire

Table C.1 lists the sentences and the answer options used to design the Portuguese version of the questtionnaire[1] used to collect data about imprecise temporal expressions. A total of 125 questions are organised in five forms (Forms A-E), each form comprising 25 questions. For each question, the target imprecise temporal expression to be evaluated is underlined.

[1] http://staffwww.dcs.shef.ac.uk/people/H.Tissot/quiz/Portuguese/

Table C.1: Questions used to design the Portuguese questionnaire.

Form A		
Question	Sentences	Answer Options
E018	última reclamação no ano passado ... usou Zolpiden por cerca de 15 dias e melhorou ... terminou um relacionamento de 4 anos recentemente ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 dias
E045	refere depressão há mais de 15 anos e que usa anti-depressivos ... consegue ficar poucos dias c/ dose menor da medicação ... humor fica oscilando, hoje reclamou de desconforto ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E028	saiu do emprego em março ... abstinente do alcool a quase 04 meses, fazendo uso de fluoxetina ... próxima consulta marcada para dia 25/02 ...	1, 2, 3, 4, 5, 6 meses
E106	exames laboratoriais solicitados em Abril ... não está conseguindo dormir bem há meses ... está há cerca de 15 dias sem as medicações ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 meses
E120	refere episódio depressivo há 2 anos ... tratada com fluoxetina, por alguns meses, com grande melhora ... com piora neste ano - refere uso de óculos ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E124	prescrito vastatina 20mg, sustrate 10 12/12h, arimidex 1mg ... dor na planta do pé E há algumas semanas ... com piora após andar muito ontem - ainda com dor no ombro ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 semanas
E044	os sintomas começaram em abril deste ano e não faz ligação com ... paciente refere que há cerca de 10 anos teve um episódio depressivo ... durante um período de 6 meses ...	5, 6, 7, 8, 9, 11, 12, 13, 14, 15 anos
E114	refere estar acima do peso há 5 anos ... teve dor precordial forte há poucos meses, ansiedade, dorme bem ... refere dor na coluna desde ontem - fez uso de dorflex ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E023	está em uso desde maio do ano passado ... conseguiu ficar mais de dois meses sem medicação ... mais 90 dias em perícia por avaliação ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 semanas
E004	último exame realizado em janeiro deste ano ... sintomas retornaram ha mais ou menos 30 dias ... estava sem usar a medicação há 4 meses ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 dias
E047	paciente relata que em Março foi submetida à cirurgia ... mas menos de 30 dias após voltou a sentir dor ... está há 1 mês sem tomar Levotiroxina ...	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 dias
E034	DUM - 12/06/2012 - faz uso de ACO ... cauterização de ferida há mais de 10 anos ... menarca aos 14 anos ...	8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18-20, 21-25 anos
E079	tratamento ambulatorial há 3 anos em uso de fluoxetina ... iniciou com 20mg/dia e há +- 18 meses já utiliza 60mg/dia ... novo encaminhamento do dia 18/08/2013 quando falou ao médico que ...	12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24 meses
E014	cirurgia ginecológica há mais de 2 anos ... mais recentemente, caso de óbito na família ... nega outros sintomas ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E100	consulta amanhã com psicóloga ... há cerca de 4 semanas foi demitido do trabalho ... estava há 4 meses trabalhando ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 dias
E029	a partir da última semana de fevereiro que estava com alteração ... diz que ficava muitos dias sem dormir e que ... ficou internado por quase 1 mês ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E025	parou de menstruar há 4 anos ... mamografia há menos de 1 ano - não trouxe o resultado ... sem uso TRH ...	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 meses
E039	diagnosticado em 2005 ... há vários anos, sem avaliação do endocrinologista ... trouxe exames (09/06/2012) ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E009	tratamento psicoterápico, há 2 anos ... agora, fazendo quinzenalmente ... estava indo 1 vez por semana ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E021	tomando só um nalapril de 12/12hs ... dor em ombro esquerdo ha mais de 30 dias ... apresentando dificuldade principalmente à esquerda, há cerca de 6 meses ...	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41-45, 46-50, 51-55, 56-60 dias
E073	overdose de medicamentos ha 6 anos ... tem recebido tratamento há mais de 5 anos e que alguns sintomas têm melhorado ... perda de familiares próximos recentemente ...	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 anos
E058	último preventivo em 2012 ... amamentou por até quase 3 anos - DUM: maio/2013 ... nega fatores de risco ...	18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 meses
E085	uso de colar cervical em 11/07/2007, sangramento irregular ... está sem medicação há mais de uma semana ... refere medo de ficar sozinha ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15-20, 21-25, 26-30 dias
E102	preventivo pela ultima vez em 2010 ... relata ter uma menstruação regular, durando cerca de uma semana ... DUM 11/12/2012 - utiliza o DIU de cobre ...	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15 dias
E067	paciente reclama que só conseguiu consulta em março ... dor em coxa esquerda há alguns dias e dor nas costas ... consulta com infectologista em 22 de agosto ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias

Form B

Question	Sentences	Answer Options
E017	último exame realizado em Jun/2012, sem alterações ... há cerca de 1 semana apresenta leucorréia ... com ooforectomia à D há cerca de 4 anos devido a mioma ...	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15 dias
E002	* RX COL LOMBO-SACRA (07/02/2012) = Sacralização de L5 ... coágulos diariamente há aproximadamente quatro semanas, associado a dor tipo ... um episódio de síncope ontem pela manhã ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 dias
E015	foi orientada em fevereiro a trocar por SERTRALINA ... medicamento em falta até recentemente ... sendo hoje novamente prescrito ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E024	depressão e ansiedade há 8 anos ... sem uso de paroxetina há mais de 2 meses ... solicita bupropiona 07-03-10 iniciou medicamento ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 semanas
E031	consulta com especialista em dezembro - atestado para INSS ... consultou há poucos dias, nova consulta agendada para ... fornecidos medicamentos para a pcte ontem(segunda-feira) ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E054	fez uso de álcool, abstinente há 5 anos ... grávida há pouco tempo (quase 6 meses de gestação) ... parou de fumar há 6 anos ...	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 meses
E005	HAS há 10 anos ... atualmente em uso de Atenolol 50mg ... atendimento de urgência ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E068	último exame em Agosto/2011 ... (leucorréia) por alguns dias ... contraceptivo hormonal. DUM = 21/10/2014, duração de 4 dias ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E010	atestado desde o início do ano ... refere estar sem medicação agora ... renovo atestado de 30 dias a partir desta data ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E080	último preventivo: 2007. DUM: 13/02/2009 ... dor em baixo ventre a +/- 2 meses, nega disuria ... Mamografia ha 2 anos, não costuma realizar autoexame ...	31-40, 41-50, 51-55, 56-59, 61-65, 66-70, 71-80, 81-90 dias
E103	esteve internado em Julho do ano passado ... após alta conseguiu ficar cerca de 15 dias sem usar a medicação ... já esteve tb internado no inicio deste ano ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 dias
E091	ultyra-som 27 semanas em 10/8 ... sangramento menstrual há mais de duas semanas ... uso de microvilar durante três semanas - refere novo sangramento ...	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31-45, 46-60 dias
E119	CD: Manter amotriptilina 25 2cp+ ret qdo ecg estiver pronto ... está sem medicamento ha alguns meses ... relata cefaléia nos últimos 15 anos ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E049	ter feito uso de dipirona hoje pela manhã ... perda de peso 5 Kg em menos de um mês ... hoje diz estar com diarreia ...	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 dias
E113	consulta com ortopedista há 1 ano ... refere cefaléia há vários anos - tem enxaqueca desde a adolescencia ... está com cefaléia há 1 semana ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E107	diabético há mais de 5 anos, usa Insulina ... está tentnado consultar há meses e não consegue vaga ... está há 2 dias sem usar Insulina, devido a falta da medicação ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 meses
E059	amanhã às 8h reavaliação com o psicólogo ... terapêuticas e depois ficou quase três anos sem usar ... que vem em uso ativo há aproximadamente um mês ...	18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 meses
E043	sindrome do panico, há mais de 20 anos ... retirada a medicação há cerca de 2 anos, mas paciente não ... refere que há 6 meses os sintomas pioraram ...	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 meses
E069	iniciar hoje e fazer uso contínuo da medicação ... aresentar sangramento por mais de 7 dias ... fazer pausa de 7 dias na cartela do anticoncepcional ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15-20, 21-25, 26-30 dias
E052	DUP = há mais de 30 anos ... fez histerectomia parcial há mais de 10 anos por mioma ... a primeira por 1 mês e a segunda por 3 meses ...	8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18-20, 21-25 anos
E096	menstruou depois do parto há 1 ano - não faz uso de anticoncepcional ... refere ouvido entupido a +/- 7 dias, com inicio de dor a 3 dias ... paciente queixa-se que a mais ou menos dois dias está com cefaléia ...	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15 dias
E075	preventivo gineco - DUM: 05/03/2006 - fez cauterização ... quer trocar DIU (tem quase 10 anos)/ ao ginecologista ... retirou lipoma de dorso há 6 meses ...	3, 4, 5, 6, 7, 8, 9, 10, 11, 12 anos
E108	está na perícia atualmente ... refere que sintomas surgiram há poucos anos ... consulta agendada para dia 01/07 ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E115	resultado de espirometria (15/04/12): limitação leve e fixa ... irradiando para nuca há muitos meses (sic) ... US nódulos sólidos em QSE/E ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-25, 26-30, 31-36 meses
E030	iniciada há 3 anos, já antes do topiramato ... refere que está há muitos dias sem venlafaxina e ciclobenzaprina ... há 8 dias sem topiramato ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias

Form C		
Question	Sentences	Answer Options
E109	cirurgia em 1994 (VLP) ... dor abdominal inferior intermitente há muitos anos ... há 6 meses a dor está pior ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E088	aprox 7 dias, USG TV de 2011 mostrou ovários micro... ... na 2ª gestação (IG aprox 12 semanas) ... nunca passou por avaliação anterior ...	5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20 semanas
E011	cirurgia de válvula mitral há 4 anos ... agora inflamado e coça ... consulta c/ cirurgia em 2 dias, ja agendada previamente ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E053	fez último exame há dois anos resultado normal ... refere corrimento amarelado há mais de seis meses ... É fumante três cigarros por dia desde os 15 anos ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18-21, 22-24, 25-30, 31-36 meses
E050	primeira consulta em 20-12-2012 ... usou a medicação por menos de 1 mes, mas melhorou dos sintomas ... com tossee cansaço há 2 dias ...	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 dias
E032	em acompanhamento psiqologico há 1 ano - não frequenta regularmente ... interrompeu ha poucos dias por orientação da psicologa ... fica fora de si por cerca de um dia e logo volta a lucidez ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E071	retorno com gineco para dia 03/08/2013- refaço pedido de MMG ... sentiu muita dor por alguns dias, agora não sente mais dor ... DUM: 03/07/2012 ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E022	agendada consulta com psiq para 12/04 ... relata que há mais de 1 mês não consegue dormir ... parou há quase dois meses o uso de olanzapina ...	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41-45, 46-50, 51-55, 56-60 dias
E093	ansioso e preocupado ... uso de IMIPRAMINA 225mg há mais de 4 semanas ... agendo retorno em um mês ...	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 semanas
E035	ocorreu por 5 vezes, a última em 2008 ... ficou abstinente por mais de 2 anos ... voltou a trabalhar no mesmo local de trabalho há quase dois anos ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36-40, 41-48 meses
E116	preventivo normal em abr/2012. Usg tv ... em uso do medicamento há vários meses ... aguardando encaminhamento ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E061	no peito há 5 dias, ontem começou com febre (39ºC) ... coriza nasal, tosse há mais de 10 dias, que incomoda muito ... refere que fez medidos ontem na PA que o preocuparam ...	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21-25, 26-30, 31-45, 46-60 dias
E019	prescrevo loratadina de 12/12 mg e pomada de dexametasona ... tosse seca há cerca de 7 dias, nega febre, nauseas e vômito ... DUM: 20/11/2009 ...	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15 dias
E003	cirurgia de coluna em 2012 devido a difilculdade ... aós a cirurgia permaneceu cerca de 1 ano sem dor ... há 1 ano retorna a dor lombar ...	6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18 meses
E081	cefaleia desde ontem. temp ontem 38.0 - auto medicação ... paciente apresenta há aproximadamente 2 meses dor em região lombar e. ... nosso atendimento há +- 15 dias, onde lhe foi solicitado ...	31-40, 41-50, 51-55, 56-59, 61-65, 66-70, 71-80, 81-90 dias
E083	DUM: 18/07/2014 ... Ciclos regulares c/ +/- 3 dias de fluxo menstrual ... Último parto há 2a4m ...	1, 2, 4, 5, 6, 7, 8, 9, 10 dias
E056	ficou algumas semanas em abril sem a medicação ... há quase duas semanas decidiu tomar apenas rivotril ... conta que há 4 dias começou a trabalhar ...	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 dias
E117	endoscopia laringea de 2003- normal ... histerectomia há alguns anos em uso de estrogenio ... Gastroduodenoscopia de 2004 - gastrite ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E078	tentou fazer a coleta dia 17/08/2013 ... parou o sangramento há varios dias ... sangrou aprox 1 semana,pequena quantidade ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 12-15, 15-20, 20-25, 25-30, 30-45, 45-50, 50-60, 60-70, 70-80, 80-90 dias
E006	problema psiquiátrico há 2 anos ... Atualmente com dor no quadril direito ... sem causa aparente ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E121	até meados de 2012 aguardando ... aumento da dose de 300mg para 350mg há poucas semanas ... piora dos sintomas nos ultimos 4 anos ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9-12, 13-15, 16-20, 21-25 semanas
E041	encerrou tratamento em 2010 ... iniciou com a medicação há algumnas semanas novamente ... INSS (ficou 45 dias afastada) ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 semanas
E042	trabalha há +de5 anos em loja de ... paciente nota piora do quadro há cerca de 2 anos ... tratamento c AMT, por cerca de 1 mês com piora do nervosismo ...	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 meses
E076	atorvastatina 20 1x; Vitamina B 12; Omeprazol ... cardiopatia ha quase 10 anos ... refere depressao ...	3, 4, 5, 6, 7, 8, 9, 10, 11, 12 anos
E016	acommpanhamento psiquiátrico há quase 10 anos ... mais recentemente está afastado do trabalho ... encaminho novamente para avaliação psiq ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]

Form D

Question	Sentences	Answer Options
E111	ficou sem tomar a AMT no início de janeiro, por aproximadamente 1 semana ... parou a medicação nos últimos dias ... na última semana tem tomado 2cp de Paracetamol ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16-20, 21-25, 26-30, 31-45, 46-60, 61-90, 91-120, 121-180 dias
E038	dor nas últimas 2 semanas, hoje não conseguiu trabalhar ... dor e edema de maõ D há varios meses, dificuldade de movimentação ... encaminhamento para o dia 21/8/2011 ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E087	paciente tem consulta marcada dia 16/10/2010 com psicólogo ... febre não aferida há +-3 dias e halitose ... tratou episódio de sinusite por 10 dias com AMOXICILINA 8/8h ...	1, 2, 4, 5, 6, 7, 8, 9, 10 dias
E063	realizou o último exame há 1 ano, mamografia e exames laboratoriais ... estralos em joelho E, há cerca de 5 anos, que piora a movimentação ... dor e paresia em MSD, há mais de 3 anos, a dor iniciou na época em que ...	1, 2, 3, 4, 6, 7, 8, 9, 10 anos
E086	marcada consulta com nutricionista para o dia 15/02/2009 ... realizado tratamento há quase 2 anos ... tem Dm faz 7 anos, HAS, tireóide ...	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 meses
E074	bastante ansiosa e angustiada ... está sem diazepam há alguns dias e demais medicações estão ok ... ultima ocorrência ha cerca de 1 ano ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E122	conseguiu atendimento para o dia de hoje, no entanto não compareceu ... pois está há várias semanas sem atendimento com psicólogo ... acidente com uma faca no dia 11/11, e foi socorrida por ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9-12, 13-15, 16-20, 21-25 semanas
E064	rx marcados para janeiro ... último preventivo há menos de um ano, não pegou resultado ... nuligesta - DUM: 22/06/2011 ...	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 meses
E051	em setembro do ano passado buscou tratamento médico ... medicação usada por 2 meses - Há menos de um mês foi diminuída a dose ... perdeu mais de 4kg em 7 meses ...	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 dias
E104	último preventivo em 2012, mamografia não lembra ... está separada há anos, tem 1 filho ... tratou ferida há 30 dias ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21-25, 26-30 anos
E089	último preventivo foi realizado em julho de 2009, negativo para neoplasia ... DUM: há +/- 2 semanas. CICLOS: regulares ... DURAÇÃO: 32 dias FLUXO: 7 dias ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 dias
E007	sempre fez uso contínuo da medicação ... atualmente está sem usar o medicamento ... esteve internado por 28 dias, recebeu alta semana passada ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E012	dor abdominal ... recorrente há 3 meses, porém mais intensa agora ... sudorese difusa, amenorréia há 2 semanas - irregularidade menstrual ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E040	todos os dias da semana, 3x/dia há 5 anos ... fez uso de Omeprazol há alguns anos, qdo estava mais ansiosa ... alterada dose para XX0mg /6h por 3 dias ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E033	retorno na policlínica até meados de março de 2013 ... sendo aumentado há pouco dias FLUOXETINA para 40mg/dia ... reduzido CLORPROMAZIN para 50mg de noite hoje para avaliar se pode ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E057	avaliação da paciente foi feita no dia 03/08 ... em abstinencia há mais de 2 anos, tratando com fluoxetina ... parou a medicação há um mês e meio atrás ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36-40, 41-48 meses
E062	piora do sintoma nasal ... a dor passou - durante mais de 10 dias, tomando a medicação ... encaminho para endoscopia, medico por mais 10 dias ...	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21-25, 26-30, 31-45, 46-60 dias
E082	realizou laqueadura tubária a 10 anos, mas continuou usando ACO ... parou de usar ACO a mais de 6 meses - nega dispareunia ... desconforto intestinal ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18-21, 22-24, 25-30, 31-36 meses
E110	ciclo menstrual irregular, DUM = 11/12/2013, duração de 3 dias ... refere 1x cauterização há muitos anos, antes das gestações ... amamentou os 3 filhos por mais de 6 meses cada ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E097	benzodiazepínicos, desde 2002, com um período de 2 meses sem ... refere há cerca de 1 mês sem uso de diazepan ... há 2 semanas episódios com tontura ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 dias
E094	preventivo e mamografia há dois anos ... amamentou por mais de um ano ... DUM: 01/02/2013 - faz uso de contraceptivo ...	10, 11, 12, 13, 14, 15, 16, 17, 18-20, 21-25, 26-30, 31-36 meses
E095	apresentou há mais de 10 anos surto psicótico ... fazendo tratamento por cerca de 6 meses, tendo sido suspensa a medicação há ... há cerca de 3 meses começou a apresentar sintomas de ...	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12 meses
E101	paciente diz ter retomado hoje o uso de Risperidona 3mg ... acompanhamento psicológico por cerca de 1 ano ... após cirurgia em 2013 ...	6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18 meses
E060	paciente traz exs data 19/09/2014 ... nervosa e ansiosa hà varios dias .hoistorico familiar ... paciente relata que há 1 semana está sentindo tontura ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 12-15, 15-20, 20-25, 25-30, 30-45, 45-50, 50-60, 60-70, 70-80, 80-90 dias
E072	uso de rivotril 0,5 mg -0-1/2-1/2 ... ha pouco mais de um mes parou com uso de rivotril ... agendou retorno dia 13 na Policlínica ...	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41-45, 46-50, 51-55, 56-60 dias

Form E		
Question	Sentences	Answer Options
E046	episódio de sangramento em jun/10 ... ficou quase 30 dias sangrando ... não usa preservativo ...	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 dias
E001	último CP há 6 anos ... fez MMg há cerca de 10 anos por suspeita de nódulo ... tabagista desde os 18 anos, mais de 6 cig/dia ...	5, 6, 7, 8, 9, 11, 12, 13, 14, 15 anos
E123	refere problema de ouvido há 4 anos ... refere otalgia há algumas semanas - refere cefaléia ... ranitidina 150mg 12/12hs/10dias ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 semanas
E013	fez cx joelho D no inicio do ano ... fazendo hidroterapia recentemente 1x por semana ... cuidado parcial ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E055	uso de fluoxetina até o dia 08/08, não renovou a receita ... em perícia pelo INSS há quase dois anos ... olhos inchados , hipertensa ...	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 meses
E118	benefício do INSS,há 2 anos e 4 meses ... queixa de lombalgia há vários meses com piora no ultimo mês ... tomar medicação 2x ao dia por 30 dias ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E037	cauterizou útero em outubro/2010 devido à cervicite ... metaplasia há poucos meses ... há 9 dias iniciou novo tratamento ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-18, 19-24, 25-30, 31-36 meses
E090	último preventivo em janeiro de 2008 - relata dispareunia ... infecção urinária ha mais ou menos 30 dias, os sintomas persistem ... usou Microvlar por mais de 20 anos, estava usando DIU a 4 anos ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 dias
E027	em Novembro e Dezembro/09 ... diz que ficou sangrando quase 2 meses direto ... foi ao Hospital 2 vezes - Fez BHCG em Dezembro, negativo ...	2, 3, 4, 5, 6, 7, 8, 9, 10 semanas
E112	comenta que no final do ano passado percebeu visão embaralhada ... hipoglicemia nos últimos dias ... compareceu a consulta no final de janeiro ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16-20, 21-25, 26-30, 31-45, 46-60, 61-90, 91-120, 121-180 dias
E048	desde agosto de 2011 com hipótese de diagnóstico ... uso de ansiolítico há mais de 12 meses ... tratamento desde julho de 2011 com medicamentos psiquiátricos ...	10, 11, 12, 13, 14, 15, 16, 17, 18-20, 21-25, 26-30, 31-36 meses
E099	anteriormente em 2005, relata uso de medicação para ... crises de cefaléia há aproximadamente 1 mês ... consultou há 1 semana com neurologista ...	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 dias
E026	solicitou exames: CP de setembro de 2007= gardnerela/negativo ... refere dor em ombro D há cerca de 5 anos ... piora há cerca de 2 semanas, reg anterior irradiada ...	1, 2, 3, 4, 6, 7, 8, 9, 10 anos
E036	em julho e agosto deste ano, foi medicada c/ rivotril ... ficou alguns dias afastada do trabalho ... teve nova crise em setembro, foi a vários médicos ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E020	exame marcado para o dia 14/10/2013 ... aguardando realizar exame ha mais de um ano ... exame foi realizado em 14/10/2013 na Policlínica Municipal ...	10, 11, 12, 13, 14, 15, 16, 17, 18-20, 21-25, 26-30, 31-36 meses
E077	relato de náuseas e vômito ... cólica abdominal, há muitos dias não se alimenta, insônia ... a última há 1 mês quando foi procurar atendimento ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12-15, 16-20, 21-25, 26-30, 31-45, 46-50, 51-60, 61-70, 71-80, 81-90 dias
E105	fez ligadura há mais de 20 anos ... refere dor em BV tipo cólica há anos ... dispareunia de intróito e de profundidade há 3 anos - refere constipação intestinal ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21-25, 26-30 anos
E098	endoscopia digestiva (24/08/2010): Esôfago e duodeno OK. ... apresentando palpitações há cerca de 6 meses ... ECG em 12/07/2009: SAE e DCRD ...	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12 meses
E125	insuficiência renal há 3 anos, secundária à diabetes ... dor no joelho esquerdo há vários anos (não sabe informar quando iniciou ao certo) ... tabagismo durante 5 anos, meio maço por dia ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10-12, 13-15, 16-18, 19-21, 22-25 anos
E066	segue com sintomas desde 2003 ... a última ocorrência há pouco menos de 1 ano ... diazepam 20 mg por dia ...	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 meses
E065	uso de colar cervical em 11/07/2007 ... sangramento irregular por mais de 30 dias ... está sem medicação há mais de uma semana ...	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41-45, 46-50, 51-55, 56-60 dias
E008	tosse produtiva há 3 meses ... atualmente não faz uso de tabaco ... nega febre ou sudorese ...	[alguns dias], [muitos dias], [algumas semanas], [muitas semanas], [alguns meses], [muitos meses], [alguns anos], [muitos anos]
E092	recebeu alta no dia 11/04, após 19 dias de internação ... aparente impregnação há cerca de duas semanas - solicito interconsulta ... internada no período de 21/03/2013 à 11/04/2013 ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 dias
E070	casada há mais de 20 anos ... em tto psiquiátrico há mais de 5 anos ... a filha vai completar 18 anos e tem bom vínculo com a mãe ...	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 anos
E084	desde ontem, dor de cabeça ... em MMSS D de inicio há mais de 1 semana ... refere diarreia e vomito de inicio há 1 dia - nega febre ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15-20, 21-25, 26-30 dias

Appendix D

Sentences used in the English Questionnaire

Table D.1 lists the sentences and the answer options used to design the English version of the questtionnaire[1] used to collect data about imprecise temporal expressions. A total of 150 questions were organised in ten forms (Forms A-J), each form comprising 15 questions. For each question, the target imprecise temporal expression to be evaluated is underlined.

[1] http://staffwww.dcs.shef.ac.uk/people/H.Tissot/quiz/English/

Table D.1: Questions used to design the English questionnaire.

Form A		
Question	Sentences	Answer Options
T140	Tim last saw his doctor in January 2007. ... He had an operation on his knee several months later, ... and was back playing football again in the end of 2008. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T069	The school holiday camping trip is in July this year. ... They are going to Dorset for a few weeks.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T107	Kim's grades have been low for 2 or 3 months ... and it has taken months of hard work to raise her grades.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T011	Joe has been married 3 times in the last 12 years. ... Each marriage has lasted less than 3 years. ... He plans to marry again in late September. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 months
T044	The cats gave birth in the first week of February, 2013. ... We left them to bond with their offspring for about 10 days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T112	Steven started revising three months ago ... and recently passed his first exam.	[days], [weeks], [months], [years]
T052	Dr James returned from France two months ago, ... after spending about 1 week working in Kenya.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 days
T081	On the 15th of February Bill and Jan got married, ... though it was less than 7 months since they first met.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T061	The new TV series began four months ago ... after many days of TV promotion.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T135	Peter started his exams yesterday, ... approximately 12 years after the last time he was studying ... so naturally he was nervous for months before. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T101	In 2001 we launched a new brand of tissues, ... more than 25 years after the last tissue brand was launched.	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 years
T122	In 2008 we held a charity ball. ... We will hold one again in approximately 1 month. ... We hope it will be even better next year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 55, 54, 56, 57, 58, 59, 60 days
T090	On 04/08/1996 Ravinda got married to Eddie, ... with celebrations lasting more than 5 days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T002	Mr Jones will move home on the 11th of December, ... and should be set up with broadband in less than 10 days, ... though the phone line will be installed weeks later. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 days
T098	The e-learning system has been installed for 10 months. ... Classroom training on it was available for more than 15 weeks.	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52 weeks

Form B		
Question	Sentences	Answer Options
T016	In 2004 a premature baby called Bobby was born, ... although it was more than 7 days before he could leave hospital. ... He was christened Bobby Stevens last week. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T144	Tom's car has been off the road since July last year, ... but was fixed in the last few weeks. ... It will need to have another MOT check-up in 12 months. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T105	Timothy's test grades were very low 10 months ago ... but they have improved recently.	[days], [weeks], [months], [years]
T039	There was a long winter in December, 2010. ... The town experience months of very cold weather. ... Spring was welcomed when it came at the end of February. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T065	Thomas spent six months in the HR department ... and then several months in the Sales department.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T043	Petra flew to Spain in May 2011 ... and stayed about 12 days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T097	The cost of living rose suddenly 18 months ago, ... according to data collected for more than 2 weeks.	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60 days
T047	The contract was signed in 2012. ... It lasted about 1 month.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 55, 54, 56, 57, 58, 59, 60 days
T060	It's my dad's birthday on 26/04/2013. ... It's been about 15 years since he held a big celebration.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T077	John Berio won 8 boxing matches in the last 15 years, ... and will be coming to town to celebrate in less than 15 days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 days
T007	Anna started her engineering career 6 years ago. ... It took her less than 8 months to find her first job. ... Within 1 year she was promoted to supervisor. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T086	On 12th of November a new coin was made, ... but was discontinued less than 2 years after.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 months
T136	We used to keep bears at the zoo five years ago. ... It took many days to do the paperwork alone for their leaving, ... and 12 weeks of planning beforehand. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T026	We established our centre at the end of 2003. ... However, it took more than 8 years to get the permit, ... and 18 months to build it. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T127	In 2012 there were many new websites created. ... The average time taken to create these was about 1 week. ... It is expected that this figure will double next year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 days

Form C		
Question	Sentences	Answer Options
T028	Julie became sales Manager three months ago, ... and is currently recruiting her team. ... She plans to have a full team by 6 weeks. ...	[days], [weeks], [months], [years]
T030	John has been skiing twice in the last 6 months, ... though he recently admitted that he didn't like heights. ... So he has booked a beach holiday for next year! ...	[days], [weeks], [months], [years]
T150	In 2002 limestone production greatly increased ... which continued for a few years, ... though production had been decreasing for many years prior. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T055	Apley forest lost many trees in 2009. ... It took approximately 10 weeks to re-plant 100 trees.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T048	Paul celebrated his birthday in September 2010, ... which he had planned for about 10 months before.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T089	George Pilla became mayor for the 2nd time on 03/08/2011, ... less than 20 years after taking up this post for the first time.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T115	Mary has been knitting a scarf for the last 6 months ... something which should really have taken just weeks.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T018	The new buses were launched in March this year. ... We will offer introductory fares for over 1 month. ... We will run another promotion for the buses in 6 months. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T008	In 2006 we imported some tropical plants. ... But in less than 2 weeks they were wilting badly. ... We will be more prepared next year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 days
T062	Amanda went to Paris in November last year ... and stayed there for several days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T120	In 2010 we had our annual sale. ... It lasted for around 15 days. ... Our sale will last even longer next year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T022	The film was released in the last week of 2002. ... This was more than 2 weeks after its planned date. ... It will be showing in cinemas for about 1 month. ...	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60 days
T066	Melina's grandmother came to stay on 31/03/2008 ... and stayed some months before returning to Italy.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T057	In 2007 Tim swam with dolphins in Florida. ... He has already booked his return there in about 2 years.	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T078	The theatre hosted a performance in September/2010. ... But it showed for less than 20 days due to low sales.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days

Form D		
Question	Sentences	Answer Options
T095	The application deadline was in August 2013, ... allowing you more than 10 months to complete the application.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T075	My father was a lawyer eight years ago. ... He practised for a few years before changing career.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T023	Radley Restaurants added pizzas to their menu in 2013. ... They offered these to customers at a discount for more than 3 weeks, ... and gave a free desert to those who came often in 1 month. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T141	The BigDog game was released in April this year. ... Hundreds of consumers expressed an interest some months before ... and could reserve their copy for 2 weeks before release. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T110	I set the deadline for submitting essays three weeks ago ... but I am currently still waiting for essays from 2 students.	[days], [weeks], [months], [years]
T108	The school year began in October 2010 ... but it has taken weeks for students to really settle.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T037	Amy has advised many fashion brands in the last 7 years. ... And recently took part in a major campaign for a new designer. ... Organizers hired a special nature set for 2 weeks. ...	[days], [weeks], [months], [years]
T051	Phillip started martial arts classes seven years ago, ... and became a black belt within about 14 months.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T137	A female horse gave birth last month ... although it had complications several days before the birth ... but was well recovered 1 week after the birth. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T003	I am going shopping in the sales tomorrow. ... Any unwanted items must be returned in less than 30 days, ... though I usually return most items within 1 week of purchase. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 days
T132	2013 was a great year for growing tulips. ... However, what we expect is our roses will be blooming well in approximately 6 years. ... That is why we are not able to enter them into a competition in May. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 years
T128	Mr Smith left the department in April 2010. ... The department was without a manager for about 2 weeks. ... The new manager's contract will expire in 18 months. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T083	In 29/08/2012 a Russian man went to the South Pole, ... less than 3 weeks after visiting the North Pole!	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 days
T012	I have been taking classes since August/2011. ... I will graduate in less than 4 years ... and will then have serious celebrations in August. ...	24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60 months
T045	It has taken 7 months to write my autobiography, ... and will be released in about 18 days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45 days

Form E		
Question	Sentences	Answer Options
T119	We started our sale three weeks ago. ... It will last for around 20 days. ... We expect to put another sale on next year too. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45 days
T004	The dolphins came to the zoo in Jun/12. ... They were settled nicely in less than 1 month, ... though it was 2 weeks before they were eating a normal diet. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 days
T040	Dean got his first job promotion one year ago, ... after applying for the new job weeks before. ... He will receive the result in weeks. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T126	They came to an agreement two months ago. ... The departments would merge in approximately 15 months. ... In two years we hope to see great rewards. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T063	On 17/11/2011 Beklan Rovers won their football match. ... They hope to make the premier league in some days.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T074	The Rovers won their basketball match on 02/02/2007, ... many years after their last win.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T035	In 2012 Mr Mercer began making olive oil ... and is currently one of the top sellers in the world, ... though they have been selling internationally for 18 months. ...	[days], [weeks], [months], [years]
T053	Dr James returned from France in December this year, ... after spending about 2 weeks in Paris.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T031	The number of hybrid vehicles increased this year, ... some of which run for days before needing recharging. ... The new car to be released in 7 months will run for longer. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180 days
T093	In January 2008 Anna passed a dance audition ... after practising her dance routine for more than 12 months.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T084	Filipo Chaucer was born in March, ... less than 6 weeks after his mother arrived back in her town.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 weeks
T070	7 months ago Kelly got her first dance assignment. ... It lasted for a few weeks.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T058	Sue started her business one-and-a-half years ago. ... For about 3 years now, business sales have been very high.	24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 months
T024	Mrs Jeffers graduated 20 years ago. ... She took over 6 months to secure her first job, ... but when she did, she was employed there for 10 years. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T087	We will be importing some new orchids next month. ... We expect to see hundreds of offspring in less than 5 years.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 years

Form F		
Question	Sentences	Answer Options
T149	Since 2007 business sales have been very high. ... We have now enjoyed several years of increased profits, ... which will fund our advertising for many years to come. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T103	9 years ago John injured his leg badly ... and is currently trialling a new treatment.	[days], [weeks], [months], [years]
T138	The Aptley Zoo closed 4 years ago, ... but hopes to re-open to the public in some days. ... The animals were transferred from other zoos over the last 18 months. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T079	The TV show was first aired 15 months ago, ... but in less than 1 month it had received negative ratings.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 days
T033	Jill painted her home two months earlier than Tom ... and spent weeks recycling unwanted household items. ... She last decorated two years ago. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T129	In the first week of March, 2014 it rained heavily. ... However it was very sunny approximately 4 weeks later. ... Then rain again in the next month! ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60 days
T123	The band released its first single one year ago. ... In about 2 months it will release its second album ... and hopefully a third next year! ...	30, 35, 40, 45, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 75, 80, 85, 90 days
T094	The country began importing beer 18 years ago, ... more than 18 months after the residents campaigned for it.	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 months
T099	Rivora's song was first released on 07/02/2007. ... That is more than 1 year after the original was released.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T113	The new nursery opened its doors in March this year ... but it took days for all 20 children to finally be admitted.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180 days
T009	In January 2007 I started the flexible course. ... It will last no more than 8 weeks. ... I plan to complete the next level in July. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 weeks
T133	The charity tennis match will be held in May 2012, ... approximately 4 years after the last charity tournament. ... We hope the money raised will fund the match next year. ...	36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60 months
T013	Jo graduated from college one year ago. ... Even though she did not have a job less than 7 years before, ... she currently works as an accounts assistant. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 years
T145	Jim Stevenson won a Gold medal on 16/10/2010. ... He was in the newspapers for a few weeks after, ... until another celebrity made the news on the 2nd of December! ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T046	It snowed badly in late January last year. ... Temperatures were especially low for about 25 days.	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45 days

Form G		
Question	Sentences	Answer Options
T111	We started delivering training in the middle of June/2014, ... and we have now just finished delivering to our Europe branch.	[days], [weeks], [months], [years]
T019	The latest sale ran for 2 weeks. ... The last sale we had was over 3 months ago. ... We will hold our summer sale in late August. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 months
T100	In June/2015 we observed a rise in the number of ants, ... more than 12 years after the last occurrence.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T001	Peter had an eye examination last month, ... after suffering with a squint for the last several years. ... Within days after treatment, he began to see improvements. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T134	The business was closed for the last week of 2013, ... and has been closing for Christmas for approximately 20 years. ... It will begin trading again within the first 2 weeks of January. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 years
T091	7 years ago the adventures climbed Snowdon ... and took more than 15 days to make the journey back home.	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60 days
T109	The singer released his album on 16-08-2013, ... the first album released by him for years.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T067	The new R10x motorbike was launched in 2006, ... after a few months of intense advertising.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T054	The spring sale will start in the next four weeks ... and will last about 8 weeks.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T072	The town celebrations were in January 2005. ... Residents were still discussing it many weeks later.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T117	We experienced a heat wave last month, ... which lasted about 3 days. ... Let's hope we have continued good weather next month. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 days
T106	Kevin rode his bike for the first time last week ... but it took days for him to feel confident riding it.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180 days
T049	In first two months of 2012 Bill studied intensely, ... though he had planned about 3 months of study.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 weeks
T082	The cost of living rose suddenly in December 2004, ... and had risen again in less than 10 months after this.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T010	7 years ago I broke my foot whilst dancing. ... In less than 6 months I was dancing professionally again, ... though I need to go for a check-up next month. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 months

Form H		
Question	Sentences	Answer Options
T142	Tim became 28 years old on 01/01/2008, ... a few months before his older brother became 32 years old. ... They plan to have a joint celebration in February. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T038	Judy became manager on 5th of March, last year. ... The move to her new office took days, ... but before 1 week she was ordering new furniture! ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180 days
T059	I'm going to the countryside next week ... where I've not been for about 7 years.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T147	It has been one year since my last dental check-up. ... I have been dreading going for many weeks. ... I finally got courage and made the appointment today. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T076	My son was born prematurely 17 years ago. ... He has had regular health check-ups for several years.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T085	A new species of fish was discovered last month, ... less than 1 year after another species was declared extinct.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 months
T036	The last building inspection took place in 2005, ... which means another inspection is now severely overdue. ... The admin department will arrange this next month. ...	[days], [weeks], [months], [years]
T121	The average temperature last week was 25. ... We checked the temperature for approximately 45 days. ... It is unlikely to be as hot next month. ...	30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60 days
T017	Tim Robert's film was released on 18/07/2013. ... It will show in cinemas for more than 20 days, ... even though it will be replaced with another film next month. ...	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 65, 70, 75, 80, 85, 90 days
T080	I returned the car to the garage last week ... as it had been less than 2 months since being fixed.	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70 days
T020	We launched our latest car engine on 08/03/2014. ... Our designers were working on the design for more than 8 months. ... In late March we will begin promoting and marketing the car. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T102	In 2003 the bears at our zoo were given a new enclosure ... which comes more than 17 years after the last major change.	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 years
T130	It has been 6 months since my last dental check. ... I am due for some further treatment roughly 15 weeks later. ... I will need to stay at home for 2 weeks to fully recover. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T041	It has taken 6 months for the plant to grow tall. ... This is the best it has ever grown after spending years in the shade. ... It will most likely grow better next year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T124	We announced today our company's plans. ... The company will start selling computers in approximately 3 months. ... We will promote these on our website two weeks before. ...	60, 65, 70, 75, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 105, 110, 115, 120 days

Form I

Question	Sentences	Answer Options
T025	The new cement base was built in August last year, ... taking in total over 2 years to build it, ... though the business has recently undergone modifications. ...	18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 months
T014	I was given a lot of responsibility last week at work, ... as I have almost 12 years of experience. ... By 2013 I would like to get a promotion. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 years
T116	We will be replacing the monitors starting this year ... but it may be years for all our monitors to be replaced worldwide.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T125	On 25/03/2012 the company launched its website. ... In approximately 6 months we will launch a second site. ... By next year we hope to see great benefits. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 months
T148	On 13/02/2009 Harry won the lottery, ... having started playing some years ago. ... He plans a luxury family holiday in late April. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T005	The deadline for applications was 02/07/2012. ... They said we would receive a response in less than 2 months. ... I got a response in just 4 weeks! ...	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70 days
T015	The rice festival is in February each year. ... For more than 3 days people throw rice in celebrations. ... It takes about 10 days to clear everything up fully. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T021	The school will hold student exams in June. ... These are expected to last for more than 1 week. ... Results will be released about 8 weeks later. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T042	Andy and Paul have been good friends since 2011. ... They exchanged contact details about 5 days after meeting.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T131	It snowed heavily in late January this year. ... This comes about 1 year after our last heavy snow storm ... and significant snow in early December the previous year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T146	The town had a hurricane in March 2015. ... The community pulled together for several weeks to repair the damage, ... building friendships for many years to come. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T143	Ryan bought a motorbike seven years ago, ... after many months of having lessons. ... He will take the test in 3 weeks. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T032	This time last year we had many car rentals. ... Although there have not been so many sales in the last months ... so we'll wait two months before increasing our advertising. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T064	I went to the Caribbean on holiday in July 2014. ... We stayed on each island for a few days each.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T104	10 days ago Marie was admitted to hospital ... and is now recovering well.	[days], [weeks], [months], [years]

Form J

Question	Sentences	Answer Options
T114	The new bears arrived at the zoo in 2002 ... but it took months for them to settle in their new surroundings.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T088	On Jul/2011 Timothy graduated from university, ... less than 15 years after completing his stage 1 high school exams.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 years
T029	Ann planted some flowers in February this year. ... They are now blossoming well. ... Hopefully they will continue to produce flowers next month. ...	[days], [weeks], [months], [years]
T073	The botanic gardens opened in May this year. ... Owners had applied for funding for it some years previously.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 years
T092	In 2005 we started our promotional campaign ... which lasted more than 45 days.	40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 65, 70, 75, 80, 85, 90 days
T071	In 2007 we grew very good crops. ... For several weeks we celebrated our good fortune.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 weeks
T056	Jerry insured his sports car last month. ... He will need to renew it in approximately 1 year.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T139	We've won the award for best coffee since 2009. ... This year's winners will be announced within the next few days ... and receive their certificate some weeks later. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 days
T068	Jan and Tim bought their current home in 2001, ... after many months of legal paperwork.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 months
T096	We started the course at the beginning of 2013, ... more than 1 week after the course administration office opened.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 days
T027	I went for a physiotherapy check-up two years ago. ... It was to check up on an injury I had over 20 years ago. ... My knee is currently giving me trouble again. ...	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 years
T118	In December last year we started trading. ... Within approximately 7 days we had launched our website. ... We plan to add more features to the website next year. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 days
T034	The new transport system was introduced in 2010 ... after years of complaints from cyclists. ... Experts took 4 months to devise a successful solution. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 years
T006	Tim moved into his city centre flat on 03/07/1995. ... But in less than 5 months he had moved out again! ... The flat has had a damp problem for at least 2 years. ...	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months
T050	The tennis tournament ended last week, ... approximately 7 months after contestants began training for it.	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 months